User Modeling in the Social Semantic Web

vorgelegt von
Dipl. Inform.
Till Plumbaum
geb. in Berlin

von der Fakultät IV - Elektrotechnik und Informatik -
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Odej Kao
Berichter: Prof. Dr. Dr. h.c. Sahin Albayrak
Berichter: Prof. Dr. rer. nat. Dr. h.c. mult. Wolfgang Wahlster
Berichter: Dr. Frank Hopfgartner

Tag der wissenschaftlichen Aussprache: 3. Dezember 2015

Berlin 2015

D 83

User Modeling in the Social Semantic Web

Thesis for the Degree Dr.-Ing.
Till Plumbaum

Fakultät Elektrotechnik und Informatik
TECHNISCHE UNIVERSITÄT BERLIN
Berlin Germany, 2015

User Modeling in the Social Semantic Web
TILL PLUMBAUM

Examiner: Prof. Dr. Dr. h.c. Sahin Albayrak
Examiner: Prof. Dr. rer. nat. Dr. h.c. mult. Wolfgang Wahlster
Examiner: Dr. Frank Hopfgartner

Fakultät Elektrotechnik und Informatik
Technische Universität Berlin / DAI-Labor
Sekretariat TEL 14
Ernst-Reuter-Platz 7
D-10587 Berlin
Telephone + 49 (0)30-314 74000/02

Printed at Berlin, Germany, 2015

"Finally!"
Star-Lord, Guardians of the Galaxy

User Modeling in the Social Semantic Web

Till Plumbaum

Berlin 2015

Technische Universität Berlin
DAI-Labor
Fakultät Elektrotechnik und Informatik
Sekretariat TEL 14
Ernst-Reuter-Platz 7
D-10587 Berlin
Telephone +49 (0)30-314 74000/02

Abstract

With the rise of the Web in the 1990's, people got access to a yet unknown amount of information, finding themselves in the role of consumers of information. Since then, information on the Web has grown exponentially. All kinds of information - good, bad, incorrect, outdated or spam - can be found on the Web. With the availability of web-enabled mobile phones, people got ubiquitous access to this information wealth. This ubiquity of information access in our everyday life offers not only new opportunities but comes with more and more challenges to deal with. Finding relevant information becomes more and more difficult. This effect is known as the *Information Overload* problem. The problem describes the fact that humans have cognitive limits to process information. To much information makes it hard to understand a topic and to make decisions. While there are tools to support users in finding information, e.g., search engines, filtering for the relevant information is still a task for every user individually.

Today, new technologies and approaches are needed to overcome the *Information Overload* problem and to support users in finding the way through all available information and deliver only the information needed. A promising approach is the application of adaptive systems. Adaptive systems, in a broader scope, are systems that help users to satisfy their information need by adapting the system and/or the displayed information to specific user requirements and therefore reducing the *Information Overload* problem. An adaptive system can be divided into three main layers:

- The data acquisition layer: In the data acquisition layer, all available information about a user is collected.
- The representation and data mining layer: The data collected in the previous layer is processed and used to build a user profile.

- The adaptation layer: The adaptation layer applies the user profile to adapt the application to the user needs.

In this thesis, we examine how the "Semantic Web" and the "Social Web" can enhance adaptive systems with the goal to solve the *Information Overload* problem. The Social Web, represented by applications such as Facebook or Twitter, enables users to express their needs and preferences. The Semantic Web provides techniques allowing to manage data in machine readable form. In this work, we develop methods and models to collect and process data from the Social Web to enhance personalization. Semantic Web technologies are used to manage the data and allow us to use it application independent. Thereby, data can be used across different applications and thus more knowledge about the user is available. The developed methods and models are applied and demonstrated in three online systems, which were developed throughout this thesis.

Zusammenfassung

Mit dem Beginn des Internetzeitalters, Anfang der 90er Jahre, stieg die Menge an verfügbaren Informationen sprunghaft an, und steigt seitdem exponentiell weiter. Dabei sind alle Arten von Informationen vorhanden - wichtige, unwichtige, richtige, falsche oder auch veraltete. Mit der Verfügbarkeit von internetfähigen Mobiltelefonen sind diese Informationen nun auch rund um die Uhr und überall verfügbar. Diese allgegenwärtige Verfügbarkeit hat allerdings nicht nur Vorteile. Das Finden von relevanten Informationen wird immer schwieriger. Man spricht dabei auch vom *Information Overload* Problem. Das *Information Overload* Problem beschreibt die Problematik, das Menschen nur begrenzte kognitive Fähigkeiten haben, um Informationen zu verarbeiten. Bei zu vielen Informationen kann dann der Mensch diese nicht mehr verstehen und Entscheidungen treffen. Es gibt zwar Anwendungen, die das Finden von Informationen unterstützen, z.B. Suchmaschinen, aber das Filtern nach relevanten Informationen obliegt dabei immer noch den Nutzern.

Um das Problem des *Information Overload*s zu lösen, unterstützen Anwendungen den Nutzer mit Personalisierungsmechanismen. Systeme, die sich an die Präferenzen des Nutzers anpassen, um dessen Informationsbedürfnis zu befriedigen, nennt man Adaptive Systeme. Ein adaptives System wird im Allgemeinen in drei Schichten unterteilt:

- Die Daten-Aggregationsschicht: In dieser Schicht werden Daten über den Nutzer gesammelt, für den eine Anwendung personalisiert werden soll.
- Die Repräsentations- und Analyseschicht: In dieser Ebene werden die gesammelten Daten des Nutzers verwaltet und aufbereitet. Es wird aus den gesammelten Daten ein Benutzerprofil mit den Präferenzen

und Abneigungen des Nutzers erstellt.

- Die Adaptionsschicht: Diese Schicht repräsentiert die eigentliche Personalisierung einer Anwendung. Basierend auf dem erstellten Benutzerprofil wird eine Anwendung, inhaltlich oder in der Visualisierung, an die Präferenzen des Nutzers angepasst.

Im Rahmen dieser Arbeit wird untersucht, wie adaptive System durch das „Social Web“ und das „Semantic Web“ verbessert werden können, um das Problem des *Information Overload*s zu lösen. Das „Social Web“, repräsentiert durch Anwendungen wie Facebook oder Twitter, erlaubt es Nutzern, eigene Interessen und Präferenzen auszudrücken. Das „Semantic Web“ bietet Technologien, die es erlauben, Daten maschinenlesbar zu verwalten. In dieser Arbeit werden Modelle und Methoden eingeführt, die Daten aus dem Social Web verarbeiten, um eine verbesserte Personalisierung zu ermöglichen. Dabei werden die Daten aus dem Social Web mittels Technologien des Semantic Web verwaltet und sind applikationsübergreifend verwendbar. Dadurch stehen mehr Daten über den Nutzer zur Verfügung, was eine bessere Personalisierung erlaubt. Diese Modelle und Methoden werden in drei Onlinesystemen demonstriert und evaluiert, die im Rahmen dieser Arbeit entwickelt wurden.

Acknowledgments

I would like to thank all people who helped me writing this thesis. Without their continuous support and encouragement, writing and finishing this dissertation would have been beyond my power.

First of all, I have to thank my fellow colleagues from my research group Information Retrieval and Machine Learning. Without these people, I would have not accomplished a single line. Special thanks goes out to Andreas Lommatzsch for answering all my questions about recommender systems and helping whenever needed. Frank Hopfgartner deserves the gratitude for proofreading this work and giving valuable feedback. Thanks also to the rest of CC IRML, current or former members, including (in no particular order) Sascha Narr, Alexander Korth, Tino Stelter, Ernesto W. De Luca, Jérôme Kunegis, Alain Said, Danuta Ploch, Benjamin Kille, Michael Meder, Erwin Gunadi, Brijnesh Johannes Jain, Esra Acar, Weijia Shao and Stephan Spiegel, for helpful discussion and a great atmosphere to perform research in. Gratitude also belongs to all co-authors who I had the pleasure to work and write with, giving me always new impulses and broaden my horizon. Many thanks to (again in no particular order) Michael Meder, Frank Hopfgartner, Sascha Narr, Elif Eryilmaz, Funda Klein-Ellinghaus, Anna Reeske, Erwin Gunadi, Christian Scheel, Sahin Albayrak, Benjamin Kille, Andreas Lommatzsch, Stefan W. Knoll, Ernesto W. De Luca, Alan Said, Brijnesh Johannes Jain, Katja Schulz, Martin Kurze, Songxuan Wu, Alexander Korth, Benjamin Hirsch, Baris Karatas, Torsten Schmidt, Jérôme Kunegis, Tino Stelter, Alexander Korth and Robert Wetzker. I also want to express my thankfulness for Prof. Albayrak, who gave me the chance to learn so many new thinks and the chance to write this dissertation.

Lastly, and most importantly I would like to thank my family who supported me through all ups and downs.

Papers included in the thesis

- Till Plumbaum, Tino Stelter, and Alexander Korth. Semantic web usage mining: Using semantics to understand user intentions. In UMAP 2009: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, pages 391–396, Berlin, Heidelberg, 2009. Springer-Verlag.

- Till Plumbaum, Semantically-enhanced ubiquitous user modeling, In Paul De Bra, Alfred Kobsa, and David N. Chin, editors, User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010, volume 6075 of Lecture Notes in Computer Science, pages 407-410. Springer, 2010.

- Till Plumbaum, Andreas Lommatzsch, Stefan Rudnitzki, Ernesto William De Luca, Holger Düwiger, and Sahin Albayrak. Adaptive music news recommendations based on large semantic datasets. In 1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain, 2010.

- Till Plumbaum, Andreas Lommatzsch, Ernesto William De Luca, and Sahin Albayrak. Serum: Collecting semantic user behavior for improved news recommendations. In UMAP 2011, Poster and Demo Session, Girona, Spain, 2011.

- Till Plumbaum, Katja Schulz, Martin Kurze, and Sahin Albayrak. My personal user interface: A semantic user-centric approach to manage and share user information. In HCI International 2011, 2011.

- Till Plumbaum, Songxuan Wu, Ernesto William De Luca, and Sahin Albayrak. User modeling for the social semantic web. In Proceedings of the second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation, volume 781, pages 78-89, Bonn, Germany, October 2011.

- Till Plumbaum, Andreas Lommatzsch, Ernesto William Luca, and Sahin Albayrak. Serum: Collecting semantic user behavior for improved news recommendations. In Liliana Ardissono and Tsvi Kuflik, editors, Advances in User Modeling, volume 7138 of Lecture Notes in Computer Science, pages 402–405. Springer Berlin Heidelberg, 2012.

- Till Plumbaum, Sascha Narr, Veit Schwartze, Frank Hopfgartner, and Sahin Albayrak. An intelligent health assistant for migrants. In Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare, 2013.

- Till Plumbaum, Sascha Narr, Elif Eryilmaz, Frank Hopfgartner, Funda Klein-Ellinghaus, Anna Reeske, and Sahin Albayrak. Providing multilingual access to health-related content. In Proceedings of the 25th European Medical Informatics Conference, 2014.

- Till Plumbaum, Funda Klein-Ellinghaus, Anna Reeske, Kristin Pelz and Frank Hopfgartner. Health Assistance for Immigrants. In Smart Information Systems - Computational Intelligence for Real-Life Applications, Springer, 2015.

- Till Plumbaum and Andreas Lommatzsch. Personalized Information Access Using Semantic Knowledge. In Smart Information Systems - Computational Intelligence for Real-Life Applications, Springer, 2015.

- Till Plumbaum and Benjamin Kille. Personalized Fashion Advice. In Smart Information Systems - Computational Intelligence for Real-Life Applications, Springer, 2015.

Co-Author:

- Alexander Korth and Till Plumbaum. A framework for ubiquitous user modeling. In IRI 2007. IEEE International Conference on Information Reuse and Integration, pages 291–297, August 2007.

- Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak. An architecture for smart semantic recommender applications. In 11th International Conference on Innovative Internet Community Systems, pages 105-114, Berlin, 2011.

- Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak, A linked dataverse knows better: Boosting recommendation quality using semantic knowledge. In 5th International Conference on Advances in Semantic Processing, Lisbon, Portugal, November 20-25, 2011.

- Federica Cena, Antonina Dattolo, Ernesto William Luca, Pasquale Lops, Till Plumbaum, and Julita Vassileva. Semantic adaptive social web. In Liliana Ardissono and Tsvi Kuflik, editors, Advances in User Modeling, volume 7138 of Lecture Notes in Computer Science, pages 176–180. Springer Berlin Heidelberg, 2012.

- Ernesto William De Luca, Till Plumbaum, Jerome Kunegis, Sahin Albayrak, Multilingual Ontology-based User Profile Enrichment, In 1st Workshop on the Multilingual Semantic Web 2010. Ontological Collaboration Engineering

- Stefan W. Knoll, Till Plumbaum, Ernesto W. De Luca, Livia Predoiu In Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources. IGI Global, 2012.

- Alan Said, Till Plumbaum, Ernesto William De Luca, Sahin Albayrak, A Comparison of How Demographic Data Affects Recommendation, In UMAP 2011, Poster and Demo Session; 2011

- Michael Meder, Till Plumbaum, and Frank Hopfgartner. Daiknow: A gamified enterprise bookmarking system. In Maarten Rijke, Tom Kenter, Arjen P. Vries, Cheng Xiang Zhai, Franciska Jong, Kira Radinsky, and Katja Hofmann, editors, Advances in Information Retrieval, volume 8416 of Lecture Notes in Computer Science, pages 759-762. Springer International Publishing, 2014.

- Michael Meder, Brijnesh Johannes Jain, Till Plumbaum and Frank Hopfgartner. Gamification of Workplace Activities. In Smart Information Systems - Computational Intelligence for Real-Life Applications, Springer, 2015.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

During the past decades, we have left the industrial age and entered the information age. One big driver was the invention of the World Wide Web [29], another the invention of mobile devices. In 2007, over one billion people of the world population were using the Internet regularly and over three billion people owned a mobile phone and counting [167]. This ubiquity of computer technology and information access in our every day life offers not only new opportunities, but also comes with more and more challenges to deal with.

With the rise of the Web in the 1990's, people got access to an yet unknown amount of information, finding themselves in the role of consumers of information. Since then, information on the Web has grown exponentially. All kinds of information - good, bad, incorrect, outdated, and spam - could be found on the Web leading to the *Information Overload* problem. The problem of *Information Overload* describes the fact that humans have cognitive limits to process information. Too much information makes it hard to understand a topic and to make decisions [102, 131, 187]. This led to an increasing dependence on tools like search engines for the management, finding and discovery of useful information.

> "Data is like food. A good meal is served in reasonably-sized portions from several food groups. It leaves you satisfied but not stuffed. Likewise with information, we're best served when we can partake of reasonable, useful portions, exercising discretion in what data we digest and how often we seek it out." *William*

> *Van Winkle* [207]

With the upcoming of the Web 2.0 a paradigm shift took place. The role of the people changed, from consumers to producers of information [146]. The term Web 2.0 usually describes a second generation of Web Services comprising blogs, wikis, social networking services (SNS) like Facebook[1] or Twitter[2], and APIs to access and mashup data or services. While there are no exact boundaries what is part of the Web 2.0 and what not, there is a common characteristic of all Web 2.0 applications: they support users to communicate and collaborate through the creation and sharing of information. These communication and collaboration aspects are also often referred to as the Social Web.

The Social Web enables users to easily create and share information, be it via blogs, social tagging services like delicious[3], social networks or multimedia content sharing via social photo applications like Flickr[4] or video sharing sites like Youtube[5]. All of these new possibilities led to an even more increased amount of information available and traditional technologies to cope with this information, like search engines, hit the wall. The problem of *Information Overload* becomes even more intense.

> ”There was 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days, and the pace is increasing.” *Eric Schmidt, CEO Google, Google’s 2010 Atmosphere convention*

Today, new technologies and approaches are needed to overcome the *Information Overload* problem and to support users to find their way through the information and deliver only the information needed (Selective Dissemination of Information [147]). A promising approach is the application of adaptive systems. Adaptive systems, in a broader scope, are systems that help users to satisfy their information need by adapting the system and/or the displayed information to specific user requirements and therefore reducing the *Information Overload* problem. According to Brusilovsky, Kobsa and

[1] http://facebook.com
[2] http://twitter.com
[3] http://delicious.com
[4] http://flickr.com
[5] http://youtube.com

Torre [48, 114, 204], an adaptive system can be divided into different tasks and sub-tasks. All authors identify three main tasks of an adaptive system:

- The data acquisition task - collecting information about users,
- The representation and data mining task - processing information and build a user model,
- The adaptation task - applying the user model to adapt the application.

This definition and segmentation of an adaptive system is also adapted in this work and presented in Fig. 1.1.

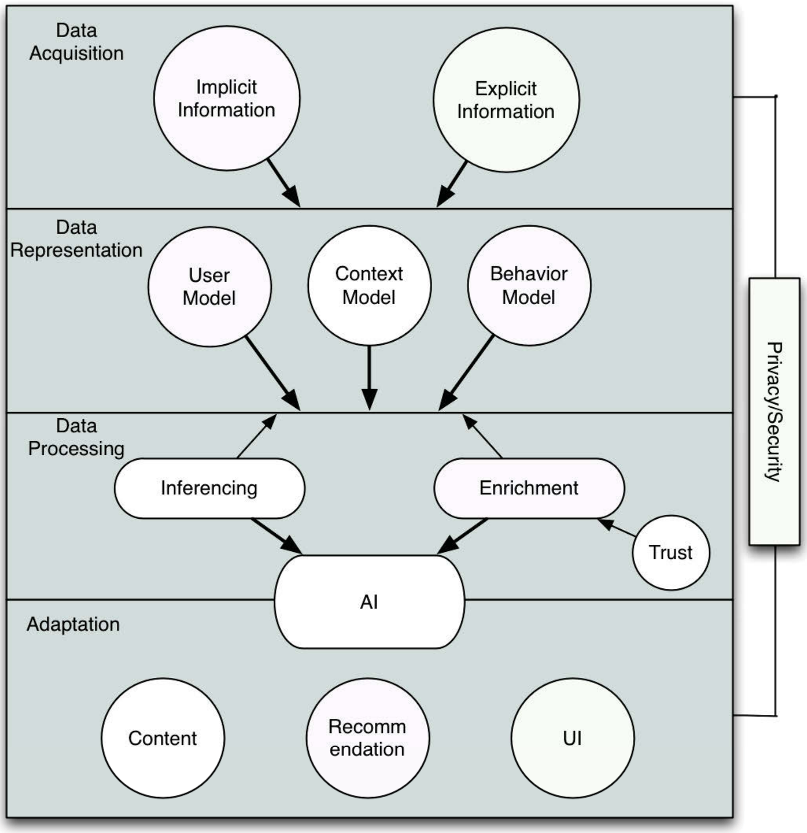


Figure 1.1: Different tasks of an *Adaptive System*.

We divided the second task into two different segments to distinguish between the work done for data representation and processing in this thesis. Well known categories of adaptive systems are personalized Information Retrieval systems (IR) and recommender systems (RS).

1.1 Motivation and Problem Description

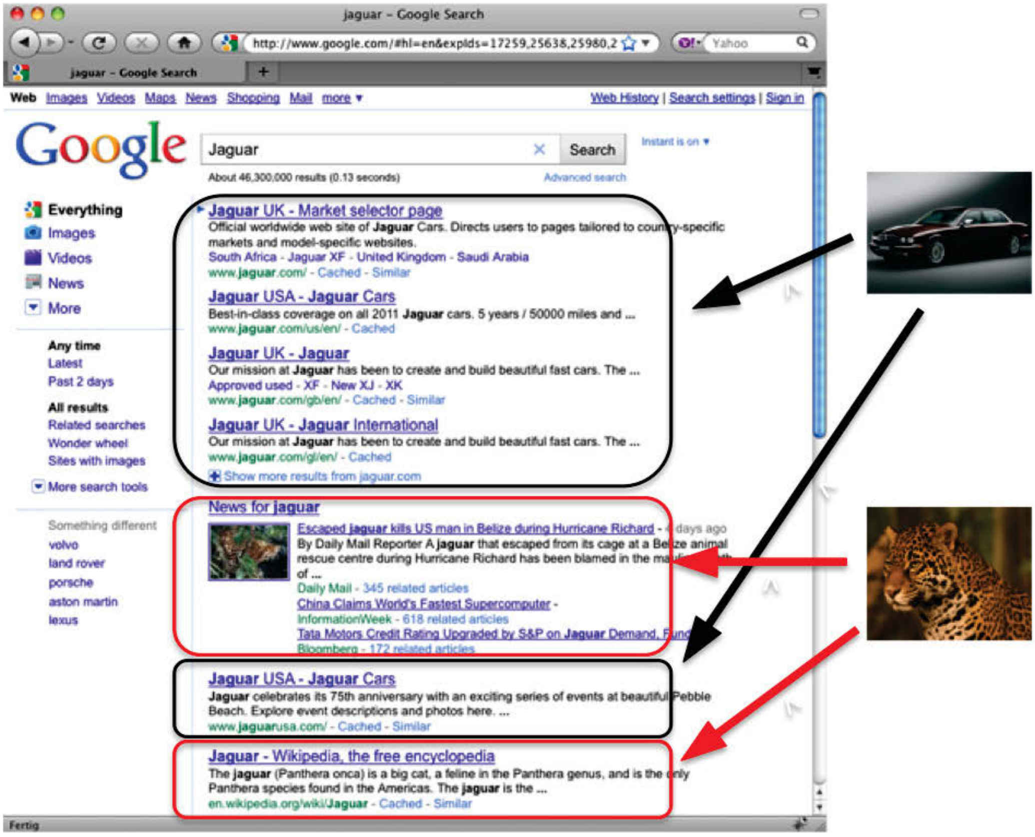
The goal of information retrieval research is providing relevant information to users without overstressing them by asking too many questions or demanding too much user participation to express their information need [22, 17]. But, in order to be able to support users, adaptive systems need as much information as possible about the user. Only with a deep understanding of the user's preferences and interests, one can deliver the information suited to the user's information need. Thereby, we hit limits of today's approaches. These limits can be classified into two problem categories:

Semantic understanding of the user's information need: Finding information in the web today usually is based on matching a given word, the information need entered by the user, with an index and returning a list of best matches. As an example, a user searches for information about *jaguar*, the animal. Therefore, the user enters the keyword *'jaguar'* into a search engine and gets back a list of results. In this example, results are returned by the English version of Google[6], see Fig. 1.2. Analyzing the result list, it becomes clear that the word *jaguar* has at least two different meanings: it can be a car or it can be an animal. Since the search engine cannot decide whether the user searched for the car or the animal, it presents both choices on the first page of the result list and leaves it to the user to decide which is the best result.

This example shows that there is a knowledge gap. While the user knows the context and background of the query, the search engine or the adaptive system does not. Thus, advanced knowledge must be added to fill this knowledge gap. The information of the user must be semantically enriched to be understandable and useful for the computer. The goal must be to have a user model containing information about interests, preferences and needs of a user where every information is clearly defined and interpretable by e.g. the search engine. We

[6] http://google.com

therefore need a semantic description, a semantic model, containing this information.



About 46,300,000 results (0.13 seconds)

Jaguar UK - Market selector page
Official worldwide web site of **Jaguar** Cars. Directs users to pages tailored to country-specific markets and model-specific websites.
South Africa - Jaguar XF - United Kingdom - Saudi Arabia
www.**jaguar**.com/ - Cached - Similar

Jaguar USA - Jaguar Cars
Best-in-class coverage on all 2011 **Jaguar** cars. 5 years / 50000 miles and ...
www.**jaguar**.com/us/en/ - Cached

Jaguar UK - Jaguar
Our mission at **Jaguar** has been to create and build beautiful fast cars. The ...
Approved used - XF - New XJ - XK
www.**jaguar**.com/gb/en/ - Cached - Similar

Jaguar UK - Jaguar International
Our mission at **Jaguar** has been to create and build beautiful fast cars. The ...
www.**jaguar**.com/gl/en/ - Cached
Show more results from jaguar.com

News for jaguar
Escaped jaguar kills US man in Belize during Hurricane Richard - days ago
By Daily Mail Reporter A **jaguar** that escaped from its cage at a Belize animal rescue centre during Hurricane Richard has been blamed in the ma of ...
Daily Mail - 345 related articles
China Claims World's Fastest Supercomputer -
InformationWeek - 618 related articles
Tata Motors Credit Rating Upgraded by S&P on Jaguar Demand, Fund
Bloomberg - 172 related articles

Jaguar USA - Jaguar Cars
Jaguar celebrates its 75th anniversary with an exciting series of events at beautiful Pebble Beach. Explore event descriptions and photos here. ...
www.**jaguar**usa.com/ - Cached - Similar

Jaguar - Wikipedia, the free encyclopedia
The **jaguar** (Panthera onca) is a big cat, a feline in the Panthera genus, and is the only Panthera species found in the Americas. The **jaguar** is the ...
en.wikipedia.org/wiki/**Jaguar** - Cached - Similar

Figure 1.2: Search results for the keyword *jaguar* using Google.

User data is application-dependent, distributed and locked up: As described before, users create and share more and more personal information, thus revealing their interests and needs in different social web applications. This distributed and heterogeneous collection of user information, stored in the proprietary user model (UM) of each application, depicted in Fig. 1.3, is a valuable source of knowledge for adaptive systems like search engines or recommender systems. The shortcomings here are that personal data is locked in the social application, as discussed in [152]. Current adaptive systems take into account user features like interest, plans and context such as the context of interaction, the device, etc. The modeling of the user is usually done in the design phase of the system, and therefore changes to the model, to adapt to changing requirements or user characteristics, can not be implemented without major changes to the system. Also the representation of the user model is in most cases strongly application-dependent and therefore not understandable and usable by other applications. That implies that the knowledge about the users, which is buried deeply in the databases of an adaptive system, can not be shared with other systems to provide better personalization and adaption results. This is known as the 'walled garden' problem [45, 71, 31] and leads to a distributed web model of a user with several partial UMs in different applications containing duplicate information.

We need to solve the heterogeneity of the user models. Current research on user model management and aggregation emphasizes two different strategies [121]. The first strategy, introduced in [27], uses a generic user model mediation framework with the focus to support recommender systems and therefore, to improve the quality of recommendations. The actual UM mediation in the framework is done by using specialized mediator components which translate the data from different models using inference and reasoning mechanisms. The second strategy focuses on the standardization of user models to allow data sharing between applications. Heckmann [90] proposes an ontological approach, the General User Model Ontology (GUMO), as a top level ontology for user models and suggest the ontology to be the standard model for user modeling tasks.

The shortcoming of the mediation layer approach is the effort needed to aggregate such heterogeneous user models. Shortcoming of standardized user models is the lack of a common standard. As long as different application providers pursue different goals and have strong own commercial interest, a global standard for a user model does not seem likely in the near future.

Concluding, to overcome the *Information Overload* problem, we need adaptive systems that tailor information to a specific user. Therefore, we need more information about the user and thus better mechanisms to interconnect user information on the Web in an inter-operable, reusable and extensible way [46].

This means that all information about users has to be described in a machine-readable interchangeable format. This allows adaptive systems to use and share information about the user, see Fig. 1.4, and to provide the best personalization possible.

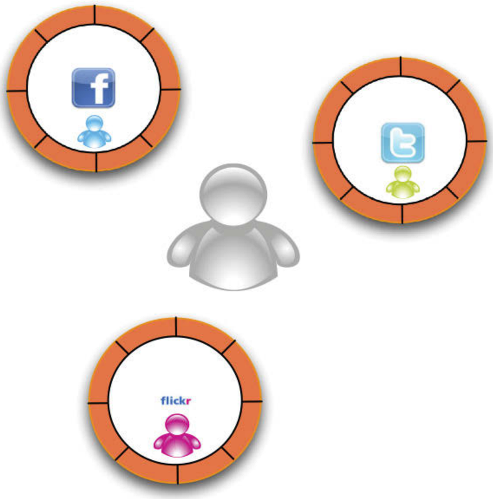


Figure 1.3: Schematic depiction of the walled garden nature of user models of three social networks of the same user.

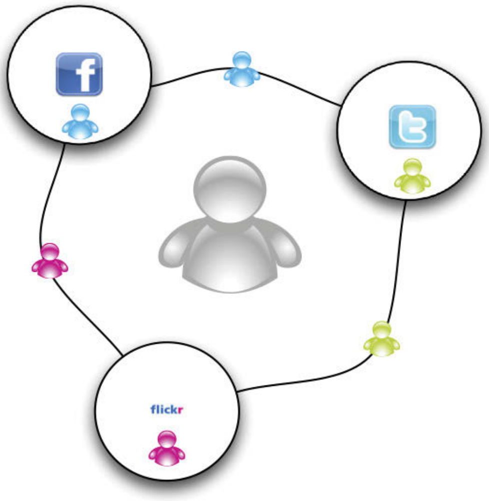


Figure 1.4: Schematic depiction of open social networks that allow to share and reuse user models and together build a holistic user model.

Creating inter-operable, reusable information is also the goal of the Semantic Web, introduced by Tim Berners-Lee [35]. The Semantic Web, basically an extension of the Web, offers methods and tools to create information that can be understood and shared between machines. Building up on those techniques (the Semantic Web and the belonging techniques are described in Section 2.2), a user information can be used to enhance adaptive systems.

1.2 Goals and Contribution

The goal of this dissertation is to enhance the personalization potentials of adaptive systems by incorporating semantic techniques into different parts of the adaptation process. Building up on the above presented general architecture of an adaptive system (see Fig. 1.1), this thesis concentrates on selected sub-parts of this process, marked yellow in Fig. 1.5, and discusses and presents methods to enhance these parts.

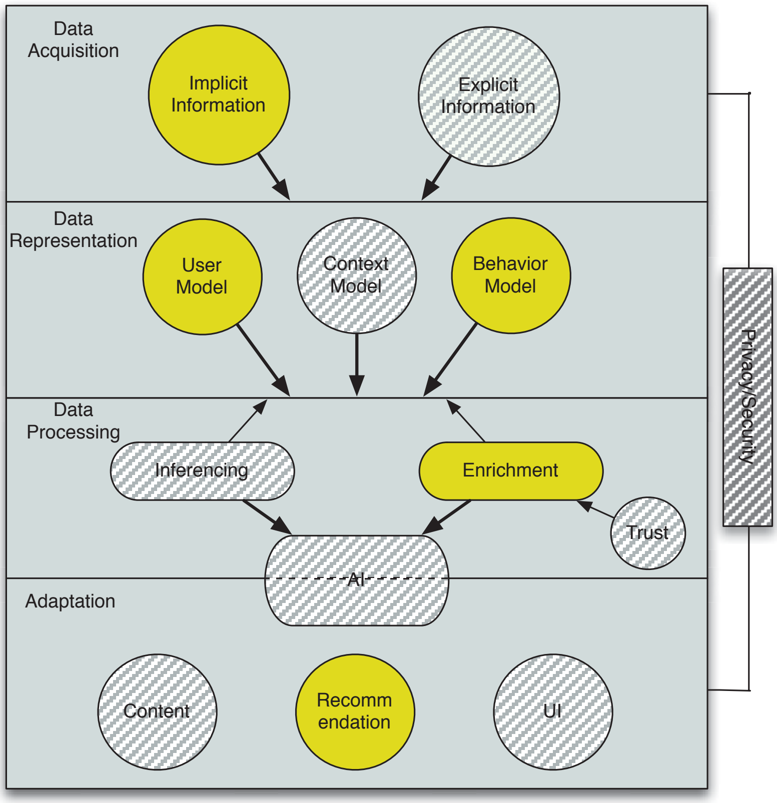


Figure 1.5: Parts of the Adaptive System that are processed in this work.

Based on the different selected sub-parts, we define research questions, which we discuss and answer in this dissertation.

Main Research Questions and Results

As motivated above, the enhancement of adaptation and personalization is crucial to cope with the increasing amount of information. This field has been approached by many research disciplines like user modeling [37, 50, 170, 27], machine learning [11, 127, 10], and human-computer interaction [104, 38] with the shared future goal to improve adaptive systems. The work presented in this thesis has been conducted from a combined Social and Semantic Web perspective. After we identified the different tasks of an adaptive system we focus on selected parts on which we align the research questions.

Data Acquisition and Representation Personal information about the user is stored everywhere on the Web as users use different applications all day. Thereby, they generate and distribute personal information like interests, preferences and goals. This distributed and heterogeneous collection of user information, stored in the user model (UM) of each application, is a valuable source of knowledge for adaptive systems. The main focus of current research is done in the area of collecting and connecting information from different services [4, 206] via standardized APIs (e.g. OAuth [87], OpenID [172]), by creating general user models [91, 47] or to aggregate information [50, 14]. Also approaches to identify users across systems [53] has been done. This thesis concentrates on two important aspects:

- User Behavior modeling and acquisition: Most information about users, their interests and needs, is collected implicitly. For a complete semantic approach, also this part has to be modeled and maintained in an ontology. Today, most approaches do not take this into account and focus only on the modeling of the user itself.
- User Models for the Social Semantic Web: Modeling users is usually an application dependent approach. Requirements, what information about users is needed, are set by the application designer. To utilize information from different applications, their is a need to have a common understanding of this data. This work takes into account the need of applications to model application dependent information and

the need for a common understanding. While general approaches like GUMO [91] try to cover all aspects of a users life, and thereby failing because they do not fulfill special needs, this work proposes a user model for the social web, specifically covering the needs of social web applications.

User Behavior Modeling and Acquisition Today, most systems depend on log file information [194] as the main source for implicit user information. In times of highly dynamic websites, this is not enough to build sufficient user profiles. Because of Java Script and new ways to render websites asynchronously, most of the user actions are not necessarily send back to the server but handled by the client.

- *How can user behavior, and the meaning of a click be collected on today's highly dynamic websites?* Not only that because of dynamic content loading not all actions are tracked, but also semantics, e.g. why does a user clicks on a link, is not captured and stored. With approaches like RDFa [6] and Microformats [56, 57], user interaction with an application can be captured already on the client-side.
- *How can semantic user behavior be modeled?* Log files only capture a small amount of user interactions, a semantic representation can cover more aspects of user behavior. A semantic model allows to manage user behavior information application-independent and thus reusable and shareable. By linking user behavior data with external knowledge, like DBpedia [39], actions can become more meaningful.

These questions will be answered in Chapter 3 where a model to manage the collected user behavior is proposed. We present a semantic user behavior model, the User Behavior Ontology (UBO), which builds a semantic model of user interactions. We show a semantic user tracking system based on RDFa and Microformats. That allows us to gain more meaningful information than only using log files. We then present a case study of a news recommendation service, using the UBO and tracking systems demonstrating how they are used to recommend more personalized news articles. These research questions align the first and second layer of the adaptive architecture.

User Models for the Social Semantic Web With the advent of the semantic web and related technologies, see Section 2.2.1, new possibilities arise to build user models that are able to manage different kind of information and allow the sharing and reuse of information. As a semantic user model describes the contained information in a machine readable way, information can be used across application borders.

- *How must a semantic user model for the social web be constructed?* The representation of the user model is in most cases strongly application-dependent and therefore not understandable and usable by other applications. To overcome walled gardens and take advantage of the available user information out there, we need to create a common understanding, by creating a common model. Thus, we need to have a model that covers all aspects of the social web, and therefore, we need to identify the requirements of the the social web for such a model.
- *How can we leverage semantic web technologies to aggregate user information from different web applications?* This implies that the knowledge about users, which is buried deeply in the databases of an adaptive system, cannot be shared with other systems to provide better personalization and adaption results. To be able to reuse such knowledge, strategies are needed to a) aggregate information from different applications and b) give information a semantic meaning.

We answer these questions in Chapter 4. In Section 4.2 we present a model to aggregate user information from different applications and a system that uses this information to build and manage an aggregated user profile. This aggregated profile is enriched with additional information from the semantic web. We present a case study which shows the aggregation process and the usage of enriched profiles for personalized recommendations. The enrichment algorithm is explained in detail in Chapter 5. In Section 4.3 we present a study of different social web applications and identify information that needs to be taken into account for a social web user model. We present such a model and a *User Model Word Net* to store, manage and aggregate user information. This model is especially adapted to the needs of the Social Web. These research questions align the second layer of the adaptive architecture.

Data Processing and Adaptation Personalization and adaptation services rely on information about the user. Recommender Systems (RS) are one major example for such services. Well known services such as Netflix or Amazon have such amount of users, and information about them, that they can use standard approaches such as collaborative filtering (CF). New services on the other hand, often have to cope with problems such as the *Cold Start Problem*, sparse user data and the *Grey sheep* problem, preventing the use of CF algorithms right from the start. Using user profiles with information from different services can help to overcome some problems, e.g. the cold start problem [137, 136]. By combining user profiles from different services and enriching them with information from the semantic web, we show that these problems can be further minimized. With the proposed semantification of user models, not only combining, reusing and sharing becomes easier, it also pushes open new doors to the knowledge of the Semantic Web. Semantic knowledge (information) is already widely available. Using this knowledge to enrich user models can extend information about user interests which can lead to better adaption results.

- *How can we leverage the growing knowledge in the Semantic Web to lower the initially needed amount of user preference data for Collaborative Filtering?* We present an generic approach (see Section 5.2) that takes existing user profile information and tries to find related knowledge in the Semantic Web to enrich the user profile with additional information that helps to improve recommendation quality.

- *How does a semantically enriched user profile influence recommendation quality?* We conducted a comprehensive evaluation of our approach, using two data sets from LastFM[7] and Facebook[8], to see how enriched user profiles affect CF recommendations.

In Chapter 5 we present a detailed evaluation how Semantic Web technologies can be used to improve adaptive systems. Focus of this work is the 'cold-start' problem and the 'grey-sheep problem'. The evaluation shows that taken into account data from the Semantic Web, initial recommendations can be improved. This research question aligns with the third and fourth layer of the adaptive architecture.

[7] http://www.last.fm/

[8] http://www.facebook.com/

1.3 Structure of the work

This work is organized as follows. The main contributions are outlined in Chapter 3-5. Chapter 3 introduces a semantic technology to capture user behavior and a model to store and manage the collected data. We also present a corresponding semantic tracking system we developed that uses the presented model and semantic technology. Chapter 4 discusses user models for the social web. We present a study on what type of information is used in the social web and present a semantic user model for the social web. We also present an approach to aggregate information from different sources using a meta model. Chapter 5 presents an evaluation focusing on enriching semantic profiles. As [136] and [4] showed, combining user profiles can already enhance the quality of adaptive services. With the advantage of semantic profiles, we show that further improvements are possible.

Chapter 6 summarizes the presented work and revisits the research questions discussed in Section 1.2. We also present an outlook for further work in the scope of this work and with the presented solutions.

Chapter 2 introduces the background knowledge needed for the understanding of the presented work in the Chapters 3-5. We define the basic terminology for the areas of user modeling and semantic web, and explain the basic techniques and approaches for recommender systems. As the different chapters cover a broad and heterogeneous set of topics, the examination with related work is done throughout the different chapters.

The following subsection presents the research contributions and their appearance in this thesis.

1.3.1 Related publications

The research contributions of this thesis have been presented at a number of peer-reviewed national and international research events. The following publications at top-level international research conferences are the core contribution to this thesis:

Chapter	Publications
Chapter 3	[165] Semantic web usage mining: Using semantics to understand user intentions. [160] Adaptive music news recommendations based on large semantic datasets. [159, 158] Serum: Collecting semantic user behavior for improved news recommendations (conference paper and extended book version).
Chapter 4	[165] Semantic web usage mining: Using semantics to understand user intentions. [164] Verbessertes Profilmanagementsystem (Patent). [152] Semantically-enhanced ubiquitous user modeling. [119] A trilogy of webs for machines. [160] Adaptive music news recommendations based on large semantic datasets [111] Collaboration ontology: Applying collaboration knowledge to a generic group support system. [64] Multilingual ontology-based user profile enrichment. [58] Semantic adaptive social web. [163] My personal user interface: A semantic user-centric approach to manage and share user information. [166] User modeling for the social semantic webs.

Chapter 5	[120] A framework for ubiquitous user modeling. [128] An architecture for smart semantic recommender applications. [129] A linked dataverse knows better: Boosting recommendation quality using semantic knowledge. [157] Personalized Information Access Using Semantic Knowledge
Chapter 6	[162] An intelligent health assistant for migrants. [161] Providing multilingual access to health-related content. [156] Health Assistance for Immigrants [155] Personalized Fashion Advice

Table 1.1: Publications to which the author contributed and their appearance in this dissertation.

Publications that indirectly contributed to the dissertation: [112, 181, 1]

Chapter 2

Background: User Modeling, Adaptation and the Semantic Web

In this chapter, we introduce the most important terms and technologies used throughout this thesis and draw connections between the single concepts to develop a common understanding. We start by giving a definition of user models and user profiles and describe what type of information is stored in such models and how it is represented. We then outline how this information is used by different adaptive systems.

2.1 Introduction to User Modeling and Adaptation

Adaptive systems, particularly user-adaptive software systems [186], tailor their behavior to specific needs and preferences of the user, i.e. reranking [144] search results based on the users interests. This system adaptation to a specific user is done on the supposition that the adaptation to different users with their different requirements increases the usefulness of such systems [192]. This adaptation effect requires that the system knows the users' needs and preferences. This information is stored in a *user model*. The process of managing and maintaining user information from adaptive systems is called *user modeling* [50].

The terms *user model* and *user modeling* are usually traced back to the works of Rich, Allen and Cohen (see [175, 9, 61]). Based on their work, numerous adaptive systems were developed that collected different kinds of user information and perform different forms of adaptation [113]. Still, the definition of the term user model (UM) and what type of information a model must contain is inconsistent among different researchers. Therefore, the next paragraphs define the term user model and how it is used throughout the thesis.

User model and *user profile* are often used as synonymous throughout literature, some authors define a user profile as a simple user model [117]. Others define 'data concerning the background, interests and general knowledge of users' as the *user profile* which is part of a user model component [24] or the 'user profile is an individual user model, a collection of information that describes the user's needs, preferences and interests' [202].

In this thesis, we make a clear distinction between user profile and user model, where the user profile is a data instance of a user model [80],.

User Model *The user model (UM) is an abstract model defining type and meaning of information stored about the user. Usually, it contains basis information like the users contact information, needs, preferences. More specialized information is domain dependent and differs from application to application.*

User Profile *The user profile is a data instance of the user model and contains the data of an specific user. The user profile information is the basis for adaption processes.*

2.1.1 A taxonomy of User Model Dimensions

Different adaptive systems have different requirements, therefore, they need different information about the user. This section presents a consolidated taxonomy about user information regarding the information needs of adaptive systems. This taxonomy is based on [192, 88, 103, 114, 50]. Table 2.1 shows the identified dimensions and which work it mentions. Before we start describing the different dimensions in detail, some general words on user model information. Not every information is equally treatable, some information is only valid for a certain time-frame or only relevant in a specific scenario. Thus, there is a distinction between different levels of modeled information:

- Short-term and long-term information: In user models, one distinguishes between information that is gathered recently, e.g. in a search context, the last queries, and information that is collected and analyzed based on data collected during a longer time period (search activity during the last months). Short-term information in adaptive system usually is used to adapt to the current context, for instance the current information need of a user. Long-term information is used to adapt to general interests of a user. In an adaptive news application short-term information could be used to determine the current information need of the user, e.g. latest election results, while the long-term information represent the user's general interest in the local football team. An example of this use is given in Section 3.4 and in [158, 37].
- Domain dependent and independent information: Independent user information is referred to as almost static personal data of a user, e.g. gender or proficiencies. Dependent information denotes dynamic properties of a user model as a result of user's interaction with a system, which usually covers user's knowledge, interests and goals.

The dimensions presented in Table 2.1 can belong to one or more of these levels. For example, interests belong to long and short-term information and are domain-dependent information.

In the following, we will describe the user information belonging to the dimensions and give some brief examples where and how the information is used.

Personal Characteristics

Personal characteristics (or demographics) span from basic information like gender or age to more social ones like relationship status. All this information

Table 2.1: Categories of user information.

Type of information	[192]	[88]	[103]	[116]	[50]
Personal Characteristics	x	x	–	x	–
Interest and Preferences	x	–	–	x	x
Needs and Goals	x	x	–	x	x
Mental and Physical State	–	x	x	x	–
Knowledge and Background	x	–	x	x	x
User Behavior	x	–	x	x	–
Context	x	–	x	–	x
Individual Traits	–	x	–	–	x

presents user specific information and represents an individual user. Information in this category is usually domain independent and changes only slowly. Such information is typically used to classify users into groups and adapt the system user interface or behavior to such groups. Adaption based on demographic stereotypes is for examples used in e-commerce systems [116], in health care systems [197] or in educational environments [106].

Interest and Preferences

User interests and preferences, which are often used as synonyms [192], are one of the most valuable inputs for adaptive systems [50]. Interests or preferences in an adaptive system usually describe the users' interest in certain items. These items can be anything, e.g. products, news or documents. Based on this knowledge about the interests, adaptive systems can tailor their service to the user. This may be the re-ranking of search results or the adaptation of a recommendation system [125]. User interests are typically represented as set of features with weights or as a list of ratings. The durability of interest information can usually be divided into two parts. Long-term interests that rarely change, like the interest in a band that one liked since the youth and short-term interests like the interest in a trending news topic.

Knowledge and Background

Besides interest and preferences, information about the knowledge and the background of the user about a topic, subject or domain is one of the most important features for adaptive systems. While interest and preferences are an indispensable source of information for e.g. recommender systems, knowledge and background (like previous academic education) is used widely in different areas of adaptive systems. It is used in adaptive educational systems to adapt the learning material to the knowledge of a student [94, 13], display personalized help texts or tailor descriptions to the technical background of a user [188]. The knowledge and background is a long-term attribute on the one hand but can differ and change from session to session depending on the topic. Knowledge and background about certain topics can increase or decrease over time [50]. The representation form of knowledge typically follows a level structure, e.g. from "novice" to "expert", "none" to "good" or 1 to 5. Information about the level can be gathered by asking the user directly or by analyzing user behavior [190, 15] .

Mental and Physical State

Mental and physical state describes individual characteristics of a user like physical limitations (ability to see, ability to walk, heartbeat, blood pressure, etc.) or mental states (under pressure, cognitive load) [88]. Such information is a valuable extension to interest and knowledge and is needed for adaptive systems like health care systems that can adapt to the users' individual state [178]. For example, interface adaptations profit from information like "ability to see" as they can tailor their output to the special needs, e.g. text to speech for blind people [95, 96, 79]. Mental and physical state values are usually long-term attributes.

Goals, Plans and Needs

When using a computer system, users usually have a goal they want to achieve. Such goals can be to satisfy an information need or to buy a product. Adaptive systems need to know that goal, understand it and know how to cope with it (referred to in the literature as plans), to be able to adapt to it. The plan to reach such goal is for example to support users by changing navigation paths [25] or to reduce the amount of information to a more relevant subset. Needs and goals are very dynamic information which can change from session to session and thus remains a main challenge for adaptive systems. The observation and interpretation of user behavior can help to understand the users' needs and goals.

User Behavior

The observation and analysis of user behavior is usually a preliminary stage to infer information for one of the previous mentioned dimensions. But, it can also serve for direct adaptation like using interaction history to adapt the user interface directly to common usage patterns of the user [25]. Also, user behavior is the most important source to retrieve implicit user information. While many systems actively ask users for their level of knowledge or interest [138], this is not always applicable or wanted. If a system requires too much interactions sometimes, users are restrained from using a system or the model of the user is not up to date [50].

We distinguish between explicit and implicit behavior. Explicit behavior means rating an item or clicking a link, which is a direct indicator for interest. The other behavior type is implicit like the time a users stays on a website could indicate the interest, a long visit time could show a stronger interest while a short period could mean the opposite. Some examples for the utilization of user behavior are task prediction or usability analysis.

Context

The definition of context is still discussed [69] and no agreement is in sight. The problem with context (in the area of adaptive systems and human computer interaction) is the change from the single user-desk-computer paradigm to ubiquitous available mobile devices, e.g. smart-phones. In this work we define context as the *who*, *what*, *where* and *when*. Who is the user or a group of users, what is the object of interest a user is interacting with, the where defines a location and the when the time of day. Other definitions can be found in Schilit et al. [184] where context is defined as location, identities of nearby people and objects, and changes to those objects. Or the definition of Abowd et al. [5] where context covers a user's emotional state, focus of attention, location and orientation, time of day, objects, and nearby people.

Individual Traits

Individual traits refer to a broad range of user features that define the user as an individual. Such features can be user characteristics like introvert or extrovert or cognitive style and learning style [177]. Individual traits are stable features that at most change slowly. The collection of those individual traits is a challenging task which often needs well-designed psychological surveys. In the area of user modeling and adaptation the focus is on cognitive style and learning style [50] to build better adaptive educational systems. An example is EDUCE [109] which builds a dynamic model of the student using Gardner's theory of Multiple Intelligences [99] to decide which resource improves the student's learning performance.

2.1.2 User Model Representations

User models contain all kinds of information, as described in the previous Section 2.1.1. Depending on the context and the intended use case different information and different requirements apply and thus, a lot of different models were researched, developed and used. In the beginning of the 1980's Rich [176] and Sleeman [191] introduced a classification scheme for user models which consists of four points:

- Conical models (stereotype-user) or a collection of models of individual users.
- Explicit models, defined by the user or system designer, or implicit models gathered from user behavior.
- Short versus long-term models.
- The nature and form of the information is contained in the user model.

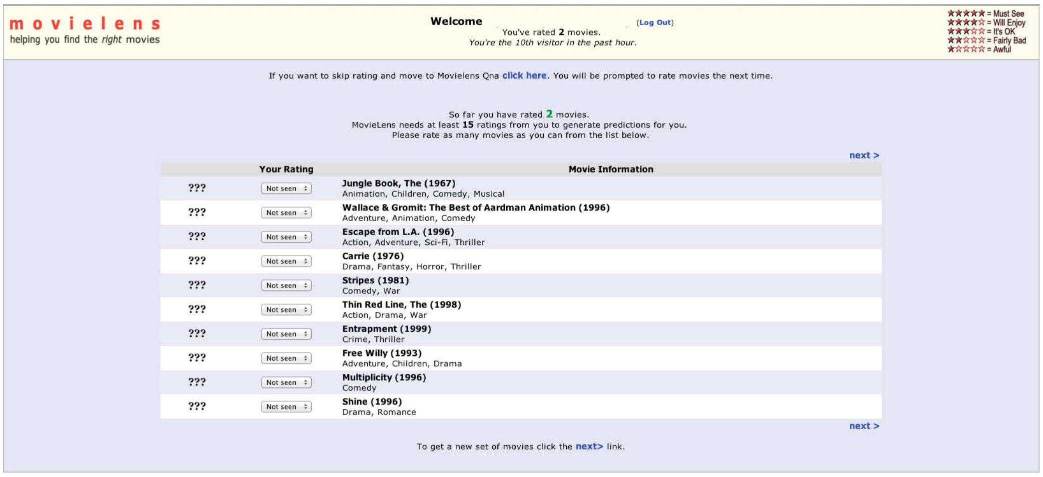
These scheme requires design decisions when designing a user model. The structure of the model has to be defined, the type of information that should be modeled and how the data is collected and maintained. This section gives a comprehensive overview of different models for different requirements. The focus lies on models for personalization and adaption purposes. One main distinction that can be made is between static and dynamic user models.

Static models are set up once and do not change over time whilst dynamic models take into account changing or new information. A simple static model is the stereotype user model [176]. The *stereotype model* defines a set of characteristics, e.g. demographic information like man or woman. Users are classified into these stereotypes and a system then adapts to these stereotypes and not to the individual user. Stereotype models are often used for new users of a system where no or little information about a user is available. Feature-based user modeling, widely used today in adaptive systems, models characteristics of individual users (preferences, knowledge, interests goals) and often also a value that indicates e.g. the level of knowledge in a certain topic. Feature-based models [50] can adapt to changing user preferences, e.g. during the users' interaction with the system, and thus the model stays up-to-date and represents the users' current state. This is important for good personalization and adaptation. Examples for feature-based models are *scalar models* and *overlay models*. A *scalar model* [191] represents a user using a single value, e.g. the knowledge of a user in a certain domain ranging from 0-5, which allows a system to adapt to the level of knowledge of a user. *Overlay models* [55] represent, similar to scalar models, the knowledge of a user for different concepts in a domain. It gives a more comprehensive view on the users' knowledge than the scalar model. Overlay models modeling users' knowledge are quite popular in adaptive hypermedia systems [50, 192], expansions of the overlay model such a the *bug model* [195] or the *genetic model* [82] were little used. Feature-based models and stereotype models are not mutually exclusive. To cope with the cold-start problem, an adaptive system can use stereotypes for the adaptation and as soon as enough data is collected switch to feature-based models.

2.1.3 Data mining and Collection

All approaches to user-based adaptation and personalization rely on sophisticated user profiles with information about the user [140, 80]. To get this information, one can distinguish between two ways: explicitly asking the user e.g. for interests and preferences or to try to learn this from implicitly gained data (or a combination of both).

Collection of explicit user information: Explicit user information collection is typically done by asking the user during the registration phase for an application. The collected information often covers demographic data, such as gender or birthday, interests and preferences.



Welcome

You've rated **2** movies.

You're the 10th visitor in the past hour.

If you want to skip rating and move to Movielens Qna **click here**. You will be prompted to rate movies the next time.

So far you have rated **2** movies.
MovieLens needs at least **15** ratings from you to generate predictions for you.
Please rate as many movies as you can from the list below.

	Your Rating	Movie Information
???	Not seen	**Jungle Book, The (1967)** Animation, Children, Comedy, Musical
???	Not seen	**Wallace & Gromit: The Best of Aardman Animation (1996)** Adventure, Animation, Comedy
???	Not seen	**Escape from L.A. (1996)** Action, Adventure, Sci-Fi, Thriller
???	Not seen	**Carrie (1976)** Drama, Fantasy, Horror, Thriller
???	Not seen	**Stripes (1981)** Comedy, War
???	Not seen	**Thin Red Line, The (1998)** Action, Drama, War
???	Not seen	**Entrapment (1999)** Crime, Thriller
???	Not seen	**Free Willy (1993)** Adventure, Children, Drama
???	Not seen	**Multiplicity (1996)** Comedy
???	Not seen	**Shine (1996)** Drama, Romance

To get a new set of movies click the **next>** link.

Figure 2.1: Movilens: Initial collection of user taste for movies.

For example, the movie recommendation service MovieLens [92], shown in Fig. 2.1, asks users to rate 15 movies before they start using the system. This initial collection allows MovieLens to compute recommendations matching the user's taste. The disadvantage with explicit feedback relies on the assistance of the user. This require extra effort from the user could restrain people to use a system. Also the given data could be incomplete as users are not willing to spend too much time entering data or the data can be outdated as the user's preferences and interests change over time.

Collection of implicit user information: Implicit user feedback collection is a non-intrusive way to get to information about the user, his interests and preferences. The advantage is that users are not forced to enter information in order to get adaptive or personalized services. However, to get meaningful information from implicit feedback takes some time and needs interaction with the system. In contrast to the explicitly collected feedback, e.g. in MovieLens, adaptation and personalization based on implicit feedback will not have the same quality in the beginning. Moreover, as the information gathered by the implicit information collection does not directly reflect the user's interests, a processing step is necessary to create a user profile from implicit information. Also the detection of negative feedback is difficult, as a click on an item or link typically reflects a positive action. In [80], a comprehensive overview of the different techniques how to collect implicit information is given. Common used sources for collecting implicit information are log files, e.g. Web logs that contain navigation behavior or search logs that contain search queries of the user.

Web Mining

One important source for implicit information is the mining of user behavior within an application. For web applications, this information is typically stored in log files, containing information about the click paths of a user. The process of gathering information from this data is called web usage mining and is part of the general web mining process. Web mining, in general, is the application of data mining methods to extract information from a webpages content, structure and usage.

Web Content Mining analysis the content, the text, on a webpage and is thus a form of text mining. With such an analysis it is possible to detect the topic, the most important terms on a website the user visited, and add those words to the user profile.

Web Structure Mining takes advantage of the semi-structured information on a webpage like the hyperlink structures. One well-known approach is the Google Page-Rank algorithm which computes a relevance score for webpages. This score, combined with words from the Content Mining, can help to weight interest in the user profile.

Web Usage Mining is, from a user modeling perspective, the most important information source as it describes the user's handling of the website. With this data, it is possible to deduce a user's interests in certain items or topics but also to detect design issues if users do not follow the click-paths as intended by the designer of an application.

In this work, the focus lies on the Web Usage Mining process and how it is influenced by the Semantic Web and the new techniques and possibilities coming along with it. This is discussed in detail in Section 3.1. In the following, we first introduce the what Semantic Web means and what new technologies exist.

2.2 Introduction to the Social and Semantic Web

Social and Semantic Web describe two different diversifications of the Web. The Social Web, or Web 2.0, created new ways for communication and collaboration over the Web. This enabled users to produce and share content and to interact and collaborate with each other. The Semantic Web, a term introduced by Tim Berners-Lee in 2001 [35], describes a Web that is enriched with semantics to allow machines to interpret information contained in it. The goal is to support data sharing and data interoperability on the Web. From a technological point of view, the Semantic Web is a set of technologies allowing sharing and understanding of information between services or machines.

Both, the Social and the Semantic Web, overlap in the goal to support sharing of information. The Semantic Web from the technological point of view and the Social Web from the user point of view. The Social Web, besides all advantages, has the problem that it is dominated by different application providers, Facebook, Twitter etc., which keep all data locked inside their applications. However, sharing and reusing of information could help to enhance personalization. With information a user entered in one application, another can personalize e.g. news [149] or Twitter streams [105]. Therefore, the goal is to enable users staying in control of their data and allowing sharing it. The Semantic Web can support this with a set of tools that allows us to describe information in a machine-readable way and thus to share and reuse data. In the following section, we will introduce these techniques. Detailed information about related work, e.g. existing approaches to semantically describe user information like FOAF, are given in the corresponding sections of the following chapters.

2.2.1 The technologies of the Semantic Web

The Semantic Web is a technology driven approach with the key challenge to ensure a common understanding of information. This can be achieved using ontologies. Ontologies are ‘an explicit specification of a conceptualization’ [84, 86]. Meaning that ontologies define a domain model describing concepts and relations in that domain. The World Wide Web Consortium

(W3C) defines a set of technologies, building on-top of each other, that combined allow to build ontologies and thus the Semantic Web. The technologies are illustrated in the Semantic Web Stack, see Fig. 2.2. The important technologies, needed in the further course of the work, are explained below.

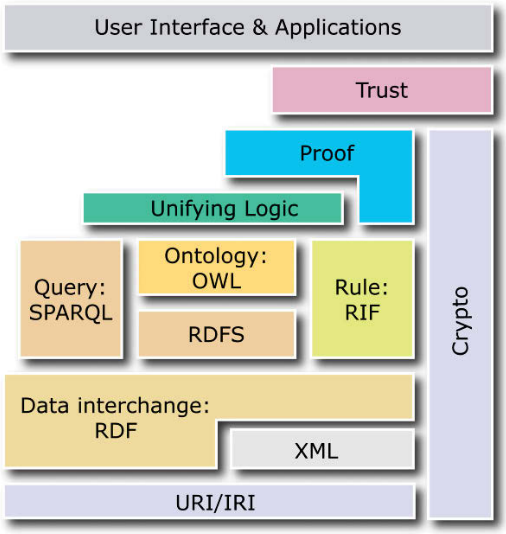


Figure 2.2: Semantic Web Stake of the W3C presented in [30]

The foundation of the Semantic Web is the Uniform Resource Identifier (URI) [34]. The URI is a unique identifier for every abstract or physical resource. A URI can denote a website (Uniform Resource Locator, URL, e.g. http://dai-labor.de) or the name of a resource (Uniform Resource Name, URN, e.g. urn:asin:B004NI3IA4 wich is a unique name for an Amazon item that could be bought an the amazon.com website). The URI enables interaction between nodes, e.g. computers, in a network to locate a resource or to unambiguously identify a resource. The Resource Description Framework (RDF) [132, 110] provides a model and a syntax to describe properties of a resource following the schema *Subject*, identifying the resource via URI, *Predicate*, defining the property, and *Object*, the value of the property. These describing subject-predicate-object triples can be coded in RDF/XML [21], Notation3 (N3, N-Triples) [32, 33] or Turtle (a subset of N3) [20] to be accessed programmatically. The SPARQL Protocol And RDF Query Language (SPARQL) [169] allows us to access and query the RDF information. These basic techniques already allow to uniquely define and describe a resource and to access this information but it does not allow to describe semantics. To describe semantics one need to be able to express relationships and hierarchies between resources. The Ontology Web Language (OWL) [134, 65], a specification of the W3C, allows exactly this. OWL allows to really build a shared understanding of a domain and its concepts in a formal language following well defined-semantics. With OWL we can define ontologies that can be shared amongst different services. Details on how semantic information is included into webpages, using RDFa and Microformats is given in Section 3.2.2.

2.3 Introduction to Recommender Systems

Recommender systems are tools to make, usually personalized, recommendations of items for customers [174, 52]. In this section we introduce the basic techniques and most used approaches of recommender systems as a prerequisite for the following chapters in this thesis.

There are many reasons to utilize recommender systems (RS) in an application or website. RS can help users finding relevant items but also help providers of RS distributing or selling more of their products. As described before, the Web brought an immense growth of available information, giving the user more choice but also introducing the *Information Overload* problem [102, 131, 187]. RS are tools that assist users by reducing the information space and thus helping them to find relevant or interesting choices.

RS research is a sub-domain of information filtering research and emerged in the 1990's as an independent field. One can divide RS in *personalized RS* and *non-personalized RS* such as most popular recommendations [173]. Non-personalized RS are a good approach if not much information about a user is available or if the computational resources are not good enough to build more complex models. Personalized recommendations, currently the focus of research, bases on explicitly expressed user preferences or inferred information by analyzing user actions. Typical knowledge sources are user preference data and demographic information, item data, user feedback in the form of ratings and interaction, and recently also context information.

2.3.1 Recommender Systems Approaches

In this section, we shortly introduce four best-known approaches for RS and their application areas.

- **Content Based Recommender Systems:** Content based RS recommend items similar to items the user liked or purchased before. Item similarity bases on the available features of compared items, e.g. recommending a crime novel because the user purchased a book from the same genre before [130, 93].

- **Collaborative Filtering Recommender Systems:** Collaborative Filtering (CF) computes recommendations for items a user may like based on previously expressed likes for different items by the user. A distinction is made between *item-based* and *user-based* CF. In item-based CF, similarities between items are computed based on previous ratings by users, and the most similar items to items the user rated are recommended [183]. User-based CF recommends items other users liked who previously liked similar items like the user the recommendations are computed for [66, 8].

- **Knowledge-Based Recommender Systems:** Knowledge-based RS combine knowledge about item attributes, domains and user preferences and need to compute if an item is useful for a user. For instance a vacation location where it is warm and affordable [74, 75]. One can distinguish between two type of knowledge-based recommenders - case-based and constraint-based recommender. Constraint-based RS try to find items that exactly match user requirements using a predefined knowledge base containing rules how to relate user requirements and items. Case-based RS on the other hand utilize similarity metrics to match user preferences with item descriptions.

- **Hybrid Recommender Systems:** Hybrid RS combine two or more different RS with the goal to overcome shortcomings of a single RS. For instance on a news recommendation website where readers have different reading preferences on weekdays and weekends [51, 52, 23].

2.3.2 Recommender Systems Problems

RS help users finding relevant items in a huge mass of items. However, no advantages without disadvantages. In real world usage, RS have several limitations and problems. CF algorithms need information about the user, so computing recommendations for new users or users with only a few expressed preferences is barely feasible. This problem is referred to as *Cold Start Problem*. Or preferences of a user do not match with the "general" expressed preferences of the majority of users. These users are called *Grey sheeps*. For these users it is hard to compute recommendations based on similarity metrics as there are not similar other users [51]. Another problem for computing similarities is the *Data Sparsity* problem. Most RS use large item sets but only have ratings for a small amount of items. Here, too, computing similarities is a problem. *Data Sparsity* and *Cold Start* are related problems.

2.4 Conclusion

This chapter provided the basic knowledge that is needed during the following chapters. It introduced the research fields the work deals with and defined the needed terminology used throughout the thesis.

We learned that different user model approaches include a lot of diverse information about a user, ranging from demographic information, over user behavior to needs and goals. We showed in Section 2.1.1 that the included information strongly depends on the task the user model is designed for. An adaptive student support system needs for instance information about the knowledge while a location based recommender needs context information. In combination with Section 4.3.1, where we analyze the requirements of the Social Web for user models, we will see that the existing approaches do not fulfill those requirements.

The presented semantic technologies OWL, RDF and SPARQL are the basis for models and approaches that are presented in the following chapters. The models introduced in Section 3.3, Section 4.3, and in Section 4.2 build on OWL and follow common semantic standards defined by the W3C. The recommender approaches presented in Section 2.3 introduce common methods for tackling the information overload problem, see Chapter 1, by recommending users a set of items personalized to the preferences of a user. We rely on these technologies as they are the standard technique, developed by the W3C[1], for the Semantic Web and fulfill all needs that come with development of a user model for Social Semantic Web.

We also introduced Recommender systems (RS) as an prominent example for adaptive systems. In this thesis, RS are used exemplarily to demonstrate the benefit that semantic user models offer. We presented typical problems that RS face in Section 2.3.2. Especially the *Cold Start Problem*, and the subordinated *Grey Sheep Problem*, are problems that could strongly benefit from user information gathered from connnected applications based on a common model. In Chapter 5 we show that not only re-using and aggregating user information helps, but also that a semantic user model allows to enrich existing user information to improve recommendation quality.

The techniques and methods presented in this chapter are important building blocks for an adaptive system that helps users to cope with the information overload problem. We showed that some of the required techniques already exist, but connections are missing to form a comprehensive adaptive system suitable for the Social Semantic Web. To adapt to user needs, adaptive systems need user information. This information currently is distributed all over the web and not re-usable. A key role to connect the different building blocks is the "semantification" of the different parts of an adaptive system, see Section 1.2. A semantic user model, fitted to the needs of the Social Semantic Web, helps re-using user information and thus enhancing the personalization process.

In the following chapter, we present an approach for semantic collection and modeling of user behavior as a first semantic building block.

[1] http://www.w3.org

Chapter 3

User Behavior and User Tracking with Semantic Technologies

In this chapter, we focus on the first and second layer of the adaptive system architecture, illustrated in Fig. 1.5, the *Data Acquisition* and *Data Representation* layer. Given background on implicit feedback and Web mining in adaptive systems from the previous chapter (see Section 2.1.3), in the next sections, we present a new approach to Web Usage Mining using semantic techniques to cope with the challenges of today's highly dynamic Web applications. We introduce two new building blocks for semantic user behavior tracking and management to get more insights about user interests and preferences based on user behavior. A deep understanding how users behave when they interact with highly dynamic and adaptive applications allows to create personalized applications and recommendation systems that support users and satisfy their needs.

Fig. 3.1 highlights (yellow bubbles) the parts that will be discussed in-depth in this chapter. We present our solution for implicit data acquisition which allows to track not only direct user actions but also semantically related information, see Section 3.2. In Section 3.3 we introduce a model that builds the basis for an application-independent management of the tracked user behavior data. Furthermore, in Section 3.4 we present a case study based on a news recommendation system we developed that combines both, the tracking system and the model, to compute a personalized news stream.

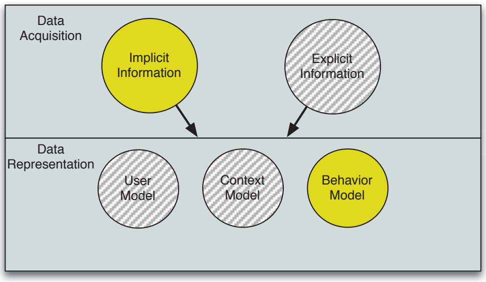


Figure 3.1: Parts of the adaptive system that are discussed in this chapter.

The main contributions for this chapter have been published in [165, 152, 159].

3.1 Introduction to Semantic Web Usage Mining

In recent years, we experienced two major paradigm shifts coming with the Web 2.0: Improved technical possibilities led to more and more complex and interactive websites and that changed the way users understand and use the Web dramatically. Today, users understand themselves as a part of the Web and demand for ways to express their opinions and thoughts. Therefore, web applications offer more and more ways to allow users to tailor the site according to their needs. Successful examples are Flickr or Facebook where users can personalize their profiles, news-feeds and share information with social contacts in several ways. These paradigm shifts, firstly from static to more complex and interactive web applications, and secondly the change of the user role from consumer to producer are accompanied with new requirements on user tracking systems and on back-end management structures.

Our goal is to extend the process of Web Usage Mining (WUM) to a Semantic Web Usage Mining process (SWUM) to collect more fine-grained data from user interactions than collected with today's tacking solutions to provide more personalized services such as recommendation and search. WUM is a sub-type of the general web mining process that focuses on the analysis of historical data such as Web server logs, browser caches, etc. to gain knowledge about the user and the way a website is used. With this information it is possible to improve a website's user interface (UI) to better match common usage patterns and to build user profiles that allow for personalization and help to meet user needs [19].

In this chapter, we present two contributions to proceed with the transition from Web Usage Mining (WUM) to Semantic Web Usage Mining (SWUM):

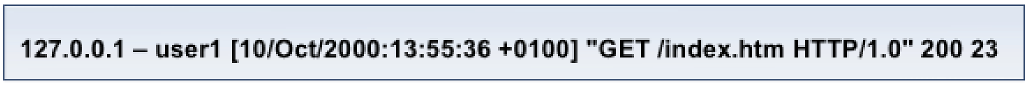
The semantic tracking of user behavior information: Before the Web 2.0, a typical Web application was a collection of HTML documents where user interaction consisted of following hyperlinks between them. Each click on a hyperlink is send to the server which in return send the requested HTML document. This form of client-server communication ensured that all user actions are recognized and processed on the server-side. In today's AJAX-based Web applications, so called Rich Internet Application (RIA), the form of client-server communication changed completely. After the initial loading of a website, the user can perform actions only on the user interface (UI) without sending data to the server. Further on, only parts of the website can be changed by requesting new information meaning that only parts of the website are changed. This can cause other data on the website to be invalid.

This new form of user interaction with a website requires new forms of user tracking systems. In Section 3.2 we introduce a new tracking approach that allows applications to track complex user interaction on the client-side.

The semantic modeling of user behavior information: The logging of user behavior has a long history, even longer than the World Wide Web. In the beginning of the 1980s, Tolle [203] started with the log analysis of the Online Computer Library Center to see to what extent different features were used by the users. One conclusion drawn from the work was that the logged data should follow well-defined requirements and have a clear structure to ease the following analysis. Till today, data collection for the web usage mining process is most often done in Web server logs [194]. A widely used log format is the Common Logfile Format (CLF), used by several web servers such as the apache web server[1]. The log file follows a predefined structure *"remotehost rfc931 authuser [date] 'request' status bytes"*, an example is shown in Fig. 3.2.

This format, while standardized and thus easy to process, still relies on

[1] http://httpd.apache.org/, Last visited 2012-05-08



```
127.0.0.1 – user1 [10/Oct/2000:13:55:36 +0100] "GET /index.htm HTTP/1.0" 200 23
```

Figure 3.2: Example of the Common Logfile Format.

the old request-response paradigm between server and client and thus did not reflect typical user interaction with today's highly dynamic webpages. In Section 3.3, we introduce a new format, a new model, to manage user behavior data in a way that fits the way how today's web applications are used. Moreover, with our new model, sharing and re-usage of collected data across applications is made possible.

3.2 Semantic User Tracking

User Tracking is the main part of the Web Usage Mining process. It allows getting detailed data about how users interact with an application. It also builds the basis for an analysis of e.g. the usability of a webpage based on detected navigation patterns or to build user profiles. These profiles are used for personalization or recommendations.

3.2.1 Background on User Tracking

Tracking of user actions can take place on both sides of the system - on the client-side or on the server-side [73]. In the following, we will present approaches for server as well as client-based solutions and discuss disadvantages and advantages of those solutions. Based on the results, we will introduce our tracking solution, overcoming disadvantages of current systems and serving as a tracking system for the future web.

Server-side tracking

In addition to the mentioned Common Logfile Format, another prominent log file example is the W3C Extended Logfile Format (ELF) which is the standard log format for Internet Information Services(IIS)[2]. The ELF provides more flexibility than the CLF as it allows customized extensions. This, on the other hand means, that such an extended version is not understandable by all applications. Such applications are so called log analyzers which process the log data to generate statistics which visualize e.g. what content is viewed by most customers, how long did they stay on a website or from which country they come from. Numerous tools exists such as AWStats[3], FastStats Log Analyzer[4] and Webalizer[5] to name a few systems available on the market today.

Research approaches such as SpeedTracer [210] und Lumberjack [60] analyze log data to identify users and to build user profiles. Lumberjack focuses on the task of grouping user sessions into common activities such as "product catalog browsing" or "financial information gathering". The user profile that Lumberjack creates is a combination of all activities of a user in a browser session that are available in the server log files. Every website is modeled as a multi-feature vector model including words, URL, in- and outlinks, describing the website and the user profile is a weighted combination of all the different vectors of the visited websites. Afterwards, the created user profile is compared with other profiles to create clusters of similar users. SpeedTracer tries to identify user sessions in log files to build distinct user profiles. The difficulty lies on the incomplete information in log files, e.g. often company servers are only visible to the outside through a single Internet Protocol address (IP address). A session is identified by IP, timestamps, URL of the requested page, referral and the browser agent. If the time between two actions is longer then a specified period, the actions will belong to two different sessions. Sessions are then analyzed to identify common paths of the user and common combinations of visited paths and pages. In Eickhoff et al. [72], the authors show that by analyzing search log-files, it is possible to learn the developing domain expertise and changing behavior of a user over time. Do-

[2] `http://www.iis.net/`, Last visited 2014-05-25

[3] AWstats is an open source log analyzer that supports all log file formats (NCSA combined/XLF/ELF log format or common/CLF log format), WebStar, IIS (W3C log format). It gives detailed information about of visits, and number of unique visitors, most visited pages etc. `http://awstats.sourceforge.net/`, Last visited 2014-05-25

[4] FastStats Log Analyzer is a commercial application that analyzes traffic patterns of a webpage to detect e.g. flaws in the design and to keep users on the site. `http://www.mach5.com/products/analyzer/index.php`, Last visited 2014-05-25

[5] Webalizer is a free log analyzer supporting the Common logfile format (CLF), several variations of the NCSA Combined logfile format), and the W3C Extended log format. The possible analyses are comparable to the other described systems. `http://webalizer.org/`, Last visited 2014-05-25

gan et al. [68] present an in-depth log analysis for the searches in the health domain, concluding that search behavior in a domain is more focused, e.g. searching explicitly for authors or genes, and users rephrasing their searches more often. Most of the presented works focus on enhancing search results. Dumais et al. [70] present a comprehensive study covering also a HCI perspective showing methods to use logs to also gain insights how an application is perceived by users.

In addition to the raw analysis of log files, some approaches uses proxy servers, which build a middle-layer between the application and the server to pickup more user data. In [16], a proxy server, the UsaProxy, is used to address the problem that log files only contain actual requests from users and not the interaction with the website, like mouse movement. The proxy server handles all HTTP requests and thus allows monitoring all visited websites. To track user interaction, the UsaProxy adds JavaScript code to every website that passes the proxy. This allows to track JavaScript events such as mouse movement, clicks etc. The focus in this work was to ease the process of usability testing by allowing a detailed tracking of user interaction, but it can also build the basis for creating a user profile.

To summarize, server-side tracking is an unobtrusive way to get data about usage patterns of websites and, to some extent, of users. A major shortcoming of server-side tracking is the fact that it only gives an incomplete picture of the user [148], which is especially bad for personalized services. Srivastava et al. [194] also mention that most browser cache pages for reoccurring requests and that these requests are not send and to that effect not logged on the server.

Client-side Tracking

Client-side tracking allows for more detailed data about the user interaction [76]. To realize client-side tracking, one can use external applications, browser extensions or approaches integrated in an application. In the following sections, we introduce some examples for these techniques and show advantages and drawbacks of client-side tracking solutions. The focus lies on current tracking solutions mainly utilizing JavaScript. Before JavaScript became a favored technique, systems often used Java applets to track user actions. Hölldobler and Michel [97] presented TELLIM, a system using Java Applets to track the user action and to generate customized multimedia presentations. Fenstermacher et al. [76] used a Python based client tracker that connects to Microsofts Internet Explorer DOM (document object model) and tracks these events.

Woopra[6] is a real-time analytics service offering a JAVA based client that allows to track and analyze several different websites. The JAVA client allows to see user behavior in real-time. To enable the tracking one has to add a JavaScript snippet into the website that should be tracked. Data is then send via AJAX to a server and from there accessible for the JAVA Client. Adding JavaScript snippets to the website is the current de-facto standard when it comes to client-side tracking. With the advent of the AJAX technology and the support by all major browsers, this technology became the preferred method. To track all elements and behavior, extra code has to be embedded on the website. One of the most prominent candidates using JavaScript snippets is Google Analytics[7].

One shortcoming of client-side systems using JavaScript is that they do not have access to referrer information of the last server requests, like what sites the user visited before. Google, the search engine part, stores this information in a cookie, which is read out by Google Analytics. Cookies are also used to recognize individual users and allow tracking across different pages. One workaround for this problem is the JavaScript History-object which contains all URLs of pages that the user has viewed in the current window of the browser. Of course this is only a subset of visited websites as users use different tabs or browser windows.

Another solution for client-side tracking without using scripts is an approach used by the etracker's Web Analytics solution[8]. Etracker loads an invisible 'counter pixel' with every page request (See Fig. 3.3) and thus the etracker server knows which pages a user has loaded.

The list of commercial client-side tracking systems can be continued endlessly. We present a few more systems that rely on JavaScript snippets to provide an insight into common characteristics: Clicky[9] tracks behavior and origin of users and offers diagrams of click activities of users. To enable Clicky to track more detailed data, specific CSS classes (comparable with the Google Analytics approach to add extra code snippets) are defined to e.g. mark

[6] `http://www.woopra.com/`, Last visited 2014-04-22

[7] `http://www.google.com/analytics/`, Last visited 2014-04-22

[8] `http://www.etracker.com`, Last visited 2014-04-22

[9] `http://getclicky.com`, Last visited 2014-05-25

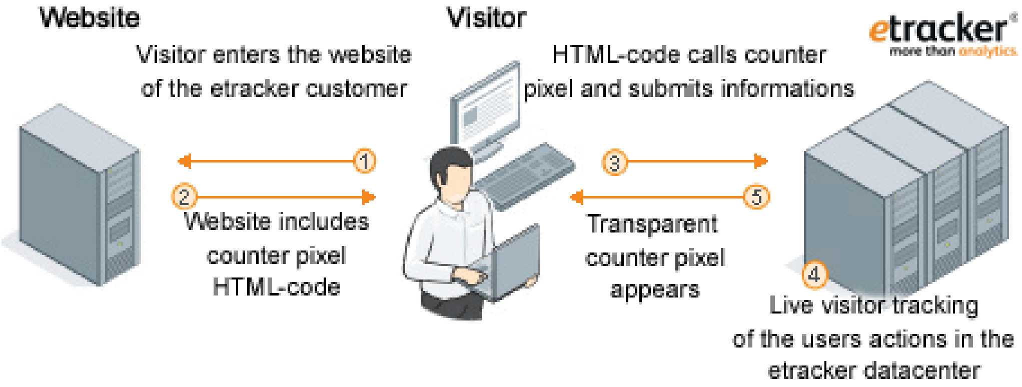

Website
Visitor
etracker
more than analytics
Visitor enters the website of the etracker customer
1
2
Website includes counter pixel HTML-code
HTML-code calls counter pixel and submits informations
3
5
Transparent counter pixel appears
4
Live visitor tracking of the users actions in the etracker datacenter

Figure 3.3: The invisible Pixel Solution which loads an invisible pixel with each page request to track the user. Picture taken from http://www.etracker.com.

elements as ignorable for the tracking component. Dojox Analytics[10] is a tracking component that focuses on developers of web applications. It allows sending error and log messages back to a server. With additional plugins, it also offers the possibility to track more user related data, e.g. where a user is active on a website. Baynote[11] is a commercial recommendation service that offers a tracking component that can be included into existing sites. Baynote tracks user behavior to learn user interests. Baynote collects data about, among other things, what a site's visitors searched for, how they move the mouse, what results are clicked on, and how long they spend looking at pages. This information is send to the server where it is analyzed. The analysis and recommendations are based on tags which are either added manually to the site itself or extracted from the search query.

Not only commercial systems take advantage of the tracking possibilities of JavaScript. Also research systems use the possibility to do a fine-grained user tracking. Mueller et al. [141] developed the system Cheese to explore if not only the mouse click but the complete mouse movement could be used to create useful user profiles. They found, that common mouse behavior across user exists and that it could be used by content providers to increase the effectiveness of their interface design. Chen et al. [59] tested if there is a correlation between mouse movement and eye movement. Experiments showed that there is a strong correlation and this could be used to improve the user interface. The research shows that the tracking of more information

[10] http://http://dojotoolkit.org/reference-guide/dojox/analytics.html

[11] http://www.baynote.com

about the user behavior helps to improve applications and thus the users' satisfaction. It can help to ease interface development, as shown in the research projects and can be used to build better user models, e.g. in the Baynote system.

To summarize, client-side tracking systems must be integrated and loaded with the webpage. Thus, it needs an active involvement of the website owner who has to integrate it. Most of these systems are powered by JavaScript. Client-side tracking allows to track more details not only about what page a user visited but also what he did on it. Server-side tracking, on the other hand, is a good and unobtrusive way to collect user behavior data but is not enough to capture all user information [83]. It collects its data through analysis of server logs and does not rely on code embedded in a webpage. A shortcoming here is that only fragmented data is available. As said, only direct page requests are logged and due to different caching strategies not even all of them. Also to mention is a privacy problem that occurs by using and sharing log data. As the examples of AOL and Netflix have shown, log files are a good way to gain knowledge about a user. But sometimes more knowledge is exploited than expected (Netflix) and anonymize data is not one-hundred percent secure (AOL) [201].

3.2.2 Approach for a Semantic User Tracking System

The presented server-side and client-side systems have shown that getting information about the user is a challenging task. Server-side tracking misses a lot of user information but is easy to use. Client-side tracking collects more data but needs to be integrated in the website itself. Nevertheless, Client-side tracking is the best way to collect user data in more detail and thus the basis for the approach presented in this section. We introduce our new tracking approach that preserves semantic knowledge found on a site while tracking user actions. Our approach extends existing solutions by broadening the tracking scope not only to user interaction on the website, but also to semantically related data belonging to an user action.

Semantic User Tracking

Capturing detailed user activity is a primary step to analyze and understand user needs. To gain meaningful information on how users interact with web applications, the collected information needs to be more detailed than that provided by tracking the navigation between pages or by analyzing web server log files. The system has to track partial reloads, clicks, mouse movement or input of text. Therefore, an advanced tracking system has to overcome the old request and response paradigm and track information to a greater degree on a JavaScript-event basis. Fig. 3.4 shows that the tracking of JavaScript-events already provides detailed information about the user interaction, e.g. it is possible to detect in which part of the page a user is active, e.g. scrolling or typing, or if he is idle and thus, allows to build a more detailed user model. On top of this, the tracked information can also easily be utilized to perform more in-depth usability tests, e.g. [59] showed that the mouse movement and the viewing direction directly corresponds to each other.

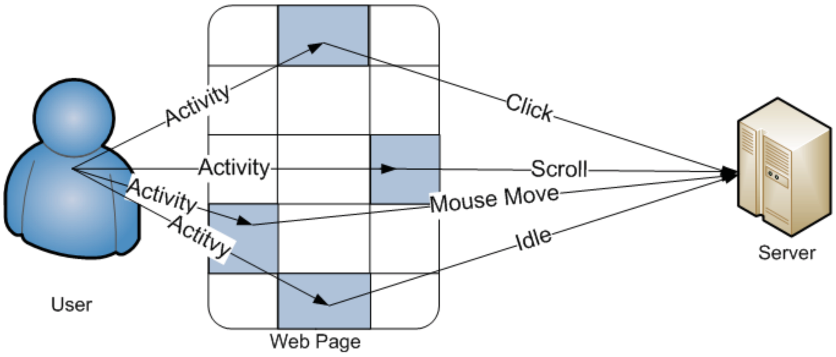


Figure 3.4: Level of detail of tracked information based on JavaScript-events.

Although JavaScript-event based tracking allows us to obtain interaction information from complex, interactive webpages, the underlying semantic knowledge and meta information about the user intention behind an action is still not captured. To overcome this drawback, our approach extends the tracking to collect meta information related to a user action. Therefore, our solution supports wide spread semantic standards like RDFa and Microformats to describe concepts on webpages. This allows us to connect interrelated information on a webpage on the one hand and to describe information on a page in detail on the other hand. Hence, it allows us to obtain more meaningful statements than by just tracking JavaScript-events. Fig. 3.5 depicts this approach. It shows interrelated information on a webpage, which could be obtained by a single user action.

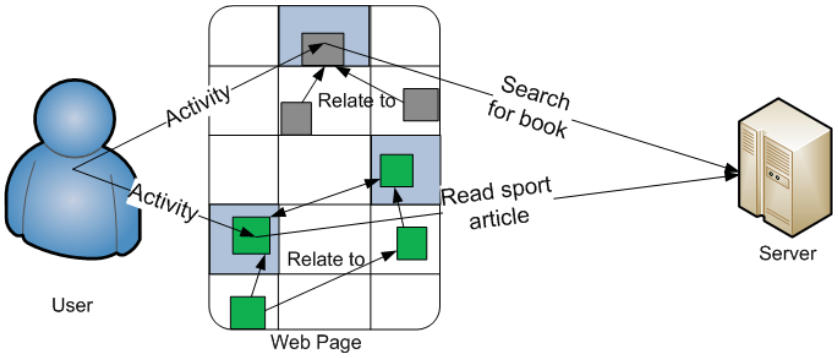


Figure 3.5: Level of detail of tracked information based on JavaScript-events enhanced with semantic information.

Requirements for Semantic Tracking

Remembering our goal, to extend the Web Usage Mining to a Semantic Web Usage Mining process, previous work and existing systems showed, that this can only be done on the client-side. While the presented systems allow to track user interaction to a great extent, it is still only the pure interaction that is tracked. To be able to deduce the user's intention or motivation behind an action, more data is needed.

This section defines the requirements for a tracking solution that is capable of the described tracking of user interactions and related data of that interaction. The requirements are a combination of requirements from different perspectives: We took into account the research view [76, 189], capabilities of current tracking systems and requirements from industry collect from the project partners during the work in SERUM [158] and PIUI [163]. The following requirements are defined:

- Detailed and accurate tracking of user actions on a page, e.g. mouse movements, scrolling, clicks or text input.
- Beside the detailed tracking of user actions, also related information on a page should be tracked. If for instance a user clicks on the second element of a result list, then the first and third results should be also send back to the server. This is important to understand why a user clicked on the second element and e.g. not on the first element.
- Unobtrusive tracking, meaning the user is not negatively affected by

the tracking.

- Customizable, the amount and type of actions that are tracked should be individually adjustable.
- Platform independence from the client operating system/browser.
- As few client-side changes as possible to use the tracking system.

The main requirement for the evolution from standard to semantic tracking is the second requirement. This says that not only the actual interaction must be tracked but also surrounding, related information which could have influenced the user action.

Semantic User Behavior Tracker

Based on the analysis of current approaches to user tracking, see Section 3.2.1, and the requirements that we defined in Section 3.2.2, the following design choices were made:

- The tracking system will be a JavaScript based client-side tracking system.
- All JavaScript events (see Fig. 3.7) can be tracked.
- Tracked information is send to a server-side counterpart that manages and saves the information.
- To gather semantic information the system supports Microformats and RDFa.

An overview of the intended system is given in Fig. 3.6.

The decision for a client-side architecture was made because of the fact, that only a client-side tracking of user interaction allows us to get the needed level of detail about user interactions. For the intended goal not only to get information about what the user does but also why, every piece of data is important. JavaScript is here the technique of choice. It is supported by all web-browsers and allows the tracking of a wide variety of different actions.

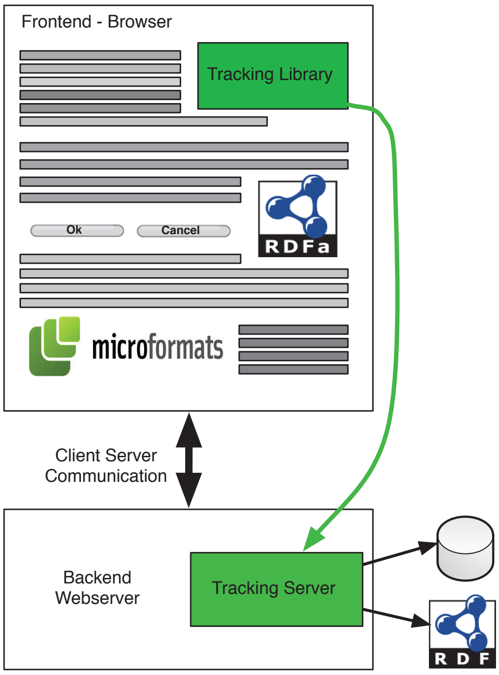


Figure 3.6: Overview of the tracking system and it's integration into existing client server architectures.

Therefore, our tracking system supports all JavaScript events as these allow direct implications on the user intention behind an action. Fig. 3.7 shows the supported JavaScript events.

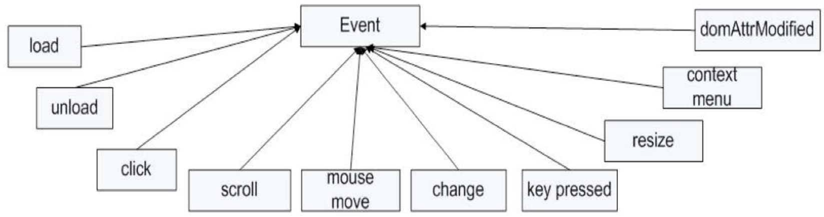


Figure 3.7: The different JavaScript-events that can be used for detailed behavior tracking.

While JavaScript based approaches are widely used in academia as well as in industry, they still lack the ability to track the semantic meaning behind user actions, as we defined in our second requirement. To track the semantics of an user action, the tracker must be able to gather additional information. This implicit information can be called context [69]. What we want to track is information that has an influence on the decisions of users and thus influences their behavior. To do so, we decided to utilize rising semantic markup languages like Microformats and RDFa. These semantic markups describe information in a semantic, machine readable way. This allows our tracker to understand what a webpage is about and what elements are on that website. Based on this understanding, the tracker is capable of tracking a users' action and related information, the context of the action.

We decided to support two markup languages initially, Microformats and RDFa, as both approaches currently have a similar distribution over the web.

Semantic markup languages help us to fulfill the second requirement, the tracking of meaning. While RDFa fully supports it, Microformats due to the missing interlinkage capabilities is only partially fulfilling it but due to its adaption in the World Wide Web an non-neglectable markup language. The tracking itself happens non-intrusive, meaning the user does not realize it. As said, RDFa is the more powerful markup language and thus the following explanations focus on *RDFa*. For Microformats see [196].

To track actual user behavior and meaning behind it, the tracking system first needs to know the semantic (ontologies), it needs to look for. This follows the Semantic Web schema, where communicating parties first need to agree on a common language and understanding. While Microformats define their own ontology, RDFa gives us the opportunity to use any available ontology needed. In the following paragraphs, we present examples for Microformats as well as RDFa, and highlight the advantages of the semantic approach.

Microformats Tracking: Microformats is an open data standard that is designed for humans first and machines second. It defines a set of simple, open data formats that extend existing and widely adopted standards such as HTML or XHTML. The goal is to use existing standards instead of building a complete new approach. Microformats intend to solve simpler problems first by adapting to current behaviors and usage patterns.

The following code shows an example of microformats. It is used in DAIKnow [135], a social bookmarking system developed at DAI-Labor. The microformats description is added to existing HTML code using the CSS-class attribute.

```
<div class="item_list_entry hproduct bookmark">
<h2 class="conversion property download">
<a href="/DAIKnow/items/details/700" class="product-title">
      Pint Labs Brews Up New Version of BreweryDB and API
</a>
<div class="product-type" style="display:none;">Website</div>
<div class="p-v" style="display:none;">
<i class="property url">http://blog.programmableweb.com/2012/04/02/
pint-labs-brews-up-new-version-of-brewerydb-and-api</i>
</div>
</h2>
</div>
```

This example shows a list entry, as part of a search list for instance, that is marked as an *hproduct*, which identifies the entry as an 'product'. By interacting with this element, our tracking system can extract information about the type (in this example a website), the url and the title. This allows to get an understanding of the object a user interacted with. Where microformats fall short is when it comes to defining relations between objects on the website.

RDFa Tracking: RDFa (Resource Description Framework - in - attributes) is a W3C Recommendation that adds a set of attribute-level extensions to

HTML for embedding rich metadata within Web documents. The RDF data-model mapping enables its use for embedding RDF subject-predicate-object expressions within XHTML documents, it also enables the extraction of RDF model triples by compliant user agents.

RDFa usage example from DAIKnow. In contrast to Microformats, RDFa uses extra attributes, such as *typeof* or *about* that are based on RDF.

```
<div typeof="ubo:Element" about="http://localhost:8080/KnowWebGui/item/700">
<span property="ubo:elementID" style="display:none">700</span>
<span property="ubo:elementType" style="display:none">Bookmark</span>
<span rel="ubo:subElementOf" resource="http://localhost:8080/KnowWebGui/item/list"/>
<div class="itemTitle float_left">
<h2 class="conversion property download">
<a href="/DAIKnow/items/details/700" class="product-title">
                 <img src="/DAIKnow/images/vw/icons/type_Website__v36991.png" />
Pint Labs Brews Up New Version of BreweryDB and API
</a>
<div>
http://blog.programmableweb.com/2012/04/02/
pint-labs-brews-up-new-version-of-brewerydb-and-api/
</div>
</h2>
</div>
```

This example shows the same entity described in the Microformats example. It defines the entity as an ubo:Element (which is introduced later on in Section 3.3) and adds several extra information, for example the type of an element, here Bookmark or an ID. This is comparable to the Microformats markup, but RDFa also allows to add links to other elements. In this example, it creates a link to a list (http://localhost:8080/KnowWebGui/item/list) and marks this element as a sub-element of this list using ubo:SubElementOf. Our tracker can follow this link and for example track what elements where displayed and what were the elements before and after the clicked entity. This can then be used to determine interests (on the clicked entity) and non-interest (e.g. on the item before the clicked one). One main advantage of this presented semantic tracking system is that it is not bound to the application. It can follow links to external semantic knowledge sources such as Freebase[12] or DBpedia[13], and extend the knowledge about the user even more. This enrichment can improve adaptation services, shown in Chapter 5. The case

[12] `http://freebase.com`, Last visited 2014-05-25

[13] `http://dbpedia.org`, Last visited 2014-05-25

study in Section 3.4 outlines a complete example from tracking, to user interests and then to recommendations. To prevent the tracker from influencing performance of an Web application, it can be configured to follow links only to certain level and to avoid cycles. The tracked information is then send to the server using the JSON format. On the server-side the information is processed and saved.

The server-side part of the system offers techniques to plug in self-defined analysis methods that can interpret the user actions. By default all user actions are stored in either a dedicated SQL-based storing solution or in an RDF-based storage system. Therefore, a component called *SemanticStore* was developed which defines Interfaces that allow to store information in SPARQL compliant storage systems. Currently, JENA[14] and Virtuoso[15] are supported. This RDF based storing is an important part as it allows us to close the gap between the semantic tracking and the storage of the information.

3.2.3 Conclusion

In the previous sections, we explained the transition from WUM to SWUM. We presented a new tracking approach, that extends current systems by using JavaScript combined with the capability to understand and track semantic markup languages (RDFa and Microformats) on the client-side to collect meaning-full information about user behavior and the underlying intention. The extension is the first step to a Semantic Web Usage mining process. It allows us to realize the tracking of the user interaction and related information on the website depicted in Fig. 3.5. The presented tracking solution is our approach for collecting implicit information task (yellow bubble) in the first layer of Fig. 1.5 - the implicit data collection.

We also build a server-side storage solution relying on RDF to store the collected data which allows us to collect semantic information on the client-side and to store it in the back-end. So far, every application still has to define their own data storage format, and thus, the analysis of the tracked data has to be adapted for each self-defined format. These self-defined formats have two drawbacks, methods and algorithms to process this data have

[14] `http://jena.apache.org`, Last visited 2014-05-25
[15] `http://virtuoso.openlinksw.com`, Last visited 2014-05-25

to be adapted for each format and collected and processed data cannot be shared to enhance personalization and recommendations of other applications. Recapitulating the advantages of the server-side logging solutions, the standardized logging formats come to mind. A standardized format to store user action captured by the client-side tracking component would close the semantic user behavior tracking gap as it makes the collected data share-able and re-usable. In the next section we introduce a new model for collecting user behavior based Semantic Web standards.

3.3 The User Behavior Model

After introducing our semantic tracking approach in the previous section and concluding that for a fully semantic tracking approach also a semantic back-end is needed, in this section we introduce a new ontology-based model for the collection and management of user behavior data, the User Behavior Ontology (UBO) [153]. The UBO serves two main goals:

1. Defining a common data model, an ontology, to manage user behavior information as described in the previous section: With UBO, data about user behavior can be collected using a common data model and thus can be shared and reused across systems. UBO defines a common schema for the semantic collection of user behavior, where the raw interaction, as well as semantic information about the user intention, context etc. can be stored. Previous work mostly focuses on domain specific modeling [171]. The data management is application independent, which means that when sharing UBO data, other applications can use the data to run their own data analysis approaches and use this for personalizing recommendations or adapting the User Interface (UI) [209].

2. Linking user behavior data with external knowledge following the Linked Open Data process: Due to the creation of an ontology as a common data model, UBO should also allow to connect behavior with external resources. This means that collected behavior can be connected to other ontologies, adding extra knowledge, for example about a user's intention behind a click, or information processed by an connected machine learning approach. This, for instance, allows to model information about what an application assumes the user is interested in, which

is valuable input for another application when data is shared.

UBO has the clear goal to serve as a general model for the interaction with an application, a semantic form of the server log files. It is not intended to be part of a general model for all possible types of behavior. The field theory of Lewin [124] proposes that human behavior is the function of both the person and the environment. With UBO, the focus is set on the environment, what type of application is the user interacting with, what elements are visible to the user etc. The user itself, their current emotions and needs are not part of UBO. This must be incorporated by other models, see Section 4.3.2.

As outlined in the previous section, the main goal of user behavior collection, the web usage mining process, is to get detailed information about how users interact with an application to find possible UI flaws and to understand what people want in order to offer better personalization or recommendation. This has to be part of UBO, too. The challenge is to build a model that allows to manage explicit information such as an click event, and to manage the implicit information that is also tracked with our tracking system.

3.3.1 Conception of UBO

UBO orients itself on the log file formats described in Section 3.2.1. As stated above, its purpose is to provide a semantic model for user behavior that can be extended with additional meta-information. Existing work on general user behavior ontologies is scarce. The work of Schmidt et al. [185] proposes a set of different models to capture all relevant data for website personalization. The used models cover the website structure (Web Portal Ontology), website content (Content Ontology), user (User Ontology), and website usage data as well as knowledge about the adaptation process itself (Adaptation Ontology). The most important ontology is the adaption ontology which is used to decide if an adaptation should take place and how to do that. The adaption decisions are based on predefined rules. The ontology most related to UBO is the Behavior Ontology [185]. The Behavior Ontology captures atomic events, such as mouse related or keyboard events, and when an event started and ended. UBO centers around the *Element* a user is interacting with. With UBO, the interaction with that a website element is tracked, how the user interacted with the element and also what other elements were visible and semantically connected. UBO allows collecting more information than the

Behavior Ontology presented in [185]. The combination of the Web Portal and the Behavior Ontology allows at least to connect an event to the page structure, but still the possibility to track underlying semantic connections on a webpage is not given. It is also not explained how the Web Portal Ontology copes with partial reloads of the website. This change in the website structure is trackable with our proposed solution, as explained in Section 3.2.2 and can be captured using UBO. Ngoc et al. [142] present an approach for generalized ontologies for user preferences, the Spatio-Temporal Ontology of User Preference (STOUP), and behavior routine, the Spatio-Temporal Ontology of User Routine (STOUR). STOUR covers part of the intended UBO functionality as it allows to model re-occuring activities in a *routine element* connected with time and system information. This is an higher-aggregation of the UBO *Event Element* but already processed to meta-knowledge. The goal of UBO is to be able to model and track atomic events and allow to process such meta-knowledge using atomic knowledge.

Before we introduce the UBO in detail, we first outline its creation and modeling process which follows a commonly agreed process for ontology creation.

Research Method

The objective when developing an ontology is to share a common understanding of the structure of information among people or software agents [84]. Today, several methods and methodologies for developing ontologies exist [62]. Uschold and Gruninger [205] present a skeletal methodology for ontology engineering. We adopted it with different methods and technologies for ontology building [85, 150]. The research approach considers the following stages:

- *Ontology goal and scope* - The scope depends on the intended usage, and the users, of the collected information - The goal of this thesis is to enhance adaptive systems e.g. for customizing intranet navigation [180]. This goal in mind, the scope of UBO is to define a model that allows us to capture all user actions within the web application, user intentions and meta-information that could be used for adaptation. UBO is supposed to support existing web usage mining approaches [26, 25] but also to support the usage of semantic information for personalization [63].
- *Ontology capture and formalization* - To define the structure of the

UBO we analyzed existing log file formats and log analyzers to see what data is logged and utilized (see Section 3.1). We incorporated knowledge from the research site [81, 182, 100, 185] and we take into account what is technical possible to track(Section 3.2.1). Based on these inputs, we define a graphical representation, see Fig. 3.16 that is used to build a conceptual model. We analyze the integration of existing ontologies to use previously established conceptualizations. The conceptual model was then transformed into a formal model and coded.

- *Ontology evaluation* - We evaluated the ontology in respect to the purpose and its intended use. In doing so, we used the test application described in Section 3.4.
- *Ontology documentation* - We documented the concepts and relationships in a data dictionary, where each concept is described by its name, description, cardinality, etc.

We used this approach to develop the UBO to capture knowledge about user behavior and intention. In the next section we present our research results in more detail.

3.3.2 Model description of UBO

UBO is a collection of different linked entities that give a complete picture of user behavior during a session and longer periods of time. It covers the users' actual behavior as well as implicit knowledge. A complete overview of UBO is given in Fig. 3.16. UBO is divided into different parts covering all aspects of the behavior life-cycle, application dependent aspects, user aspects and interaction aspects. In the remainder of this section, we describe the most important entities, their function, their properties and their intended usage.

Application Aspect

The application aspects cover all information about the application required that the user is interacting with. What type of application is it, what different views (e.g. different webpages) belong to it and what is modeled/displayed on the page.

Application

The OWL class *Application* defines the name and an ID for the application that is used to identify the application. It allows links to the *ubo:Domain* to determine the scope of the application and to the different *ubo:Views* the application has. An *Application* can consist of multiple views but must define at least one. An application can cover several domains, e.g., a news website. In this case, the different *ubo:Views* define a specific domain, e.g. sport.

Application Object Properties		
Predicate	Inverse Direction	Description
ubo:hasView	ubo:isPartOfApplication	The relation describes, that every application consists of different views (see *ubo:View* OWL-Class). An application must have at least one view.
ubo:designedForDomain	ubo:validForApplication	An application can be designed for a special domain. This relation connects the application with the *ubo:domain*.
Application Data Properties		
Predicate	Object Type	Description
ubo:applicationName	rdfs:Literal	This property defines the name of the application.
ubo:applicationID	xsd:Long	This property defines a single, unique ID for the application. In contrast to **ubo:applicationName**, this property must be unique

Table 3.1: Properties of the OWL class **Application**

View

The OWL class *View* defines a single view (e.g. webpage) of a *ubo:Application*. It can define a *ubo:Domain* (which can be different from the general application domains) and link to different *ubo:Elements*. A *View* can contain several *ubo:Element*s. Those *ubo:Element* objects can describe an entity, e.g. an artist, that is covered in an article or define a link to a different *View*.

View Object Properties

Predicate	Inverse Direction	Description
ubo:isPartOfApplication	ubo:hasView	This relation defines which application the view belongs to.
ubo:hasElement	ubo:isPartOfView	This relation specifies which elements are connected to that view.
ubo:viewHasDomain	ubo:domainUsedBy	This allows for connecting a domain to a view. E.g. this view is about the domain of "Sport"

View Data Properties

Predicate	Object Type	Description
ubo:viewName	rdfs:Literal	The name of the view. E.g. "Football news view"

Table 3.2: Properties of the OWL class **ubo:View**

Element

The OWL class *Element* marks parts of the website as contentual relevant. An *Element* can be an artist on a news page or a link to another *ubo:View* or external page. *Elements* can also refer to each other in one *ubo:View* to define that *Element*s are related. With the *ubo:elementRank* property, rankings can be defined, e.g. the rank of the element in a search result list. This is be helpful when computing an interest model as *Elements* above the element the user interacted with my not be interesting.

Element Object Properties

Predicate	Inverse Direction	Description
ubo:isPartOfView	ubo:hasElement	This relation declares which **ubo:View** an element belongs to.

ubo:relatedTo	–	An element can be related to another element. E.g. a “Submit”-button that is related to a form.
ubo:subElementOf	–	This relation pools **ubo:Element**s that are sequences or somehow sorted lists of **ubo:Element**s . E.g. a search result list. The result list itself is a **ubo:Element** and the results are **ubo:Element**s too. That means all results are **ubo:subElementOf** the search result list element.
Element Data Properties		
Predicate	Object Type	Description
ubo:elementID	rdfs:Literal	Unique ID of an element within the **ubo:Application**.
ubo:elementType	rdfs:Literal	This property defines the type of the element. E.g. button, text field, link, etc.
ubo:elementDescription	rdfs:Literal	This property can include additional information or comments describing the element.
ubo:elementRank	xsd:Integer	This property is related to **ubo:subElementOf**. To stay with the search result list example, this property denotes the rank of the element in the result list. The rank must be positive, natural number ($\mathbb{N}_1$)

Table 3.3: Properties of the OWL class **Element**

Domain

The *Domain* defines the topic of a *ubo:View* or *ubo:Application*. It allows to define a name for it, which can be a textual description. More important is the property *ubo:domainURL* which defines the domain by giving it a unique URI which is a commonly agreed description of the topic. Recommended is to use URLs from large encyclopedic resources such as Wikipedia or its semantic equivalent DBpedia. This approach, which follows the Linked Data Principles (see [41, 40]), allows other applications to understand what the application or view is about.

Domain Object Property		
Predicate	Inverse Direction	Description
ubo:validForApplication	ubo:designedForDomain	This relation connects a domain with an **ubo:Application**.
ubo:domainUsedBy	ubo:viewHasDomain	This relation connects a domain with an **ubo:View**.
Domain Data Property		
Predicate	Object Type	Description
ubo:domainTerm	rdfs:Literal	The name of the domain, e.g. “Sport”.
ubo:domainURL	rdfs:Literal	Link to a detailed description of the domain. This link follows the LOD principles, so this URL could link to the DBpedia page about sport.

Table 3.4: Properties of the OWL class **Domain**

User Aspects

User aspects include all information about the user, the device that is used to access an application and session information. This information allows to identify a user and to distinguish between different devices that a user may use. This helps to identify contexts, e.g. mobile or at home, and give better recommendations based on the context. The session information helps to narrow down the context, as it allows to unambiguously differentiate when the user did what. This allows to create context such as at work, during lunch etc.

User

The *User* entity in UBO allows to identify a user. As mentioned in Section 3.3, UBO is not focusing on the user itself, but has the goal to collect data about the user interaction and the context, the environment, of the interaction to have sophisticated data that allows for inferring interests and intention of a user. Therefore, the *User* entity only allows to set a login name, which can be a user name or ID, and to link it to a *session*.

User Object Properties		
Predicate	Inverse Direction	Description
ubo:hasSession	ubo:sessionOwnedBy	The relation connects a user with a **ubo:Session**. All events in that session are now linked to the user.
User Data Properties		
Predicate	Object Type	Description
ubo:fullName	rdfs:Literal	Name of the user.
ubo:loginName	rdfs:Literal	Login name of the user.

Table 3.5: Properties of the OWL class **User**

Device

The *Device* entity describes all relevant properties of a device, mobile, PC, etc., that helps to later distinguish between different devices of a user. That could be a notebook and PC which both run the same OS but with different screen resolutions, or a mobile device. This could be used to adapt UI elements or to determine a context, office, home or on the road.

Device Object Properties		
Predicate	Inverse Direction	Description
ubo:deviceUsedIn	ubo:hasDevice	This defines which device a user used during the session.
Device Data Properties		
Predicate	Object Type	Description
ubo:deviceOS	rdfs:Literal	Name of the OS or URI to LOD.
ubo:loginName	rdfs:Literal	Defines which input types are supported by the device.

Table 3.6: Properties of the OWL class **Device**

SessionContext

The OWL class *SessionContext* describes a time-frame when a user interacted with an application or multiple applications without a longer pause in between. It defines a start and end time and set the used devices. A *SessionContext* belongs always to one *ubo:User*.

Interaction Aspects

The interaction aspects cover all entities that help to manage the actual behavior. Information about what the user did on a webpage, read an article, clicked a link or hovered over a picture, etc., is important for later personalization and recommendation purposes. While the application aspects give us insights on how the application is structured and thus allows to draw implicit conclusions from the way the user interacted with it, the interaction gives us an explicit feedback. The click on a recommended item indicates that it matches the users' interests. To what extent depends on the further interaction, if the user for instance buys the item than it is a strong indicator for a positive perception, while a quick return to the recommendation lists indicates that the recommended item probably did not match the users' interests.

SessionContext Object Properties		
Predicate	Inverse Direction	Description
ubo:sessionOwnedBy	ubo:hasSession	This relation connects a Session with a User.
ubo:hasDevice	ubo:deviceUsedIn	This defines which device a user used during the session.
SessionContext Data Properties		
Predicate	Object Type	Description
ubo:sessionID	rdfs:Literal	Unique session id. E.g. the browser session id.
ubo:sessionBegin	xsd:DateTime	Defines the start time and date of the session.
ubo:sessionEnd	xsd:DateTime	Defines the end time and date of the session.

Table 3.7: Properties of the OWL class **SessionContext**

Event

The OWL class *Event* describes the type of event (click, mouse over, etc.) and the *Element* or *View* the user interacted with. An event always occurs on an *Element* or *View*. With the type of the *Event*, also the time when the event happened is tracked. This allows to later identify chains of actions and create higher order events. For example a click event on an element, followed by a mouse move, followed by a click release event on a different element could be a Drag-and-Drop event where an item is dropped into a basket.

Event Object Properties		
Predicate	Inverse Direction	Description
ubo:triggeredFrom	ubo:generateEvent	This relation connects an event with the user who triggered the event. This only happens if the user is known, e.g., logged in.
ubo:registeredIn	ubo:recordedEvent	This relation connects an event with a session

ubo:occuredInElement	ubo:elementUsedIn	This relation links an **ubo:Event** to an **ubo:Element** the event occured in.
ubo:occuredInView	ubo:viewUsedIn	This relation connects a domain with an **ubo:View**.
Event Data Properties		
Predicate	Object Type	Description
ubo:eventType	rdfs:Literal	The type of the event. Scroll, Click, etc.
ubo:eventTime	xsd:Time	This assigns a time value to the triggered **ubo:Event** .
ubo:eventDate	xsd:Time	This assigns a date value to the triggered **ubo:Event**.

Table 3.8: Properties of the OWL class **Event**

3.3.3 Conclusion

In this section, we presented a new ontology, the User Behavior Ontology (UBO), that provides the basic concepts and properties for describing and modeling user behavior in an application as a semantic graph. With UBO, we are able to manage and store the data collected with the tracking system, presented in Section 3.2.2, without losing information. The semantic information tracked on the client-side are stored in a one-to-one manner. In this way, we track data about the structure of an application and how the user interacted with it. This allows us to analyze User Interface(UI) lacks and what differences changes to the UI make. It also allows us to create user models about interests, needs and preferences which is needed for personalization and recommendation.

As discussed in Section 3.3.1, existing ontologies only cover parts of the UBO functionality, mainly leaving out the possibility to model and manage site structures [142] or do not allow to track underlying semantic relation between an element a user interacts with [185]. UBO also allows to manage changing site structures, e.g. through partial relaods, with is important to fully track user intentions.

UBO builds a generic semantic back-end to store and manage collected user behavior information. In combination with our tracking system a full semantic tracking process is made possible. The next section presents a prototype, that uses the tracking system and UBO to give users recommendations based on their tracked interaction with a news application service.

3.4 SERUM - Semantic Recommendations based on large unstructured datasets

How can semantic tracking and data management technologies be leveraged for personalization and recommendation services? In order to address this question, we present the SERUM system (Semantic Recommendations based on large unstructured datasets), a news recommendation system that utilizes semantic technologies to collect implicit user behavior and to build semantic user models. These models, combined with large-scale semantic data sets, are then used to compute personalized news recommendations using graph-based algorithms. We introduce the building blocks of SERUM for the semantic data management, personalization and recommendation, with the main focus on the implicit user behavior collection. Therefore, SERUM uses RDFa annotations and the RDFa tracker (see Section 3.2.2) to collect meaningful user behavior and our User Behavior Ontology (UBO, see Section 3.3) to build semantic user behavior models. In the following sections we first introduce the idea and goal of the SERUM project, afterwards we explain the architecture of the SERUM system and then present the use cases that the semantic web usage mining approach should cover and demonstrate it with an example based on the SERUM system.

3.4.1 Goal of SERUM

Finding relevant news on the Internet is becoming increasingly difficult as the number of news published every day is exploding. A search on Google

News[16] for the term 'Ebola' returned 161,000 results retrieved in one day. To master this information overload, several personalized filtering approaches have been proposed. One of the first projects was the 1992 started GroupLens project [118] that recommended Usenet news based on collected ratings from other readers. With our web-based application SERUM (Semantic Recommendations based on large unstructured datasets), we support users in finding interesting and up-to-date news about their favorite topics, currently focusing on entertainment news. Therefore, we utilize a broad range of semantic technologies to further enhance the personalization and recommendation quality. While other work focuses on only one aspect of semantic personalization support (e.g. [204]), we build a holistic semantic approach, including front- and back-end solutions, to better learn a user's interest and thus to better recommend news matching these interests. We incorporate semantic information on the client-side, using RDFa[17] in the user interface and a user-tracking component that is able to track this RDFa information [165]. In the back-end, we have a semantic knowledge base that includes information from semantic encyclopedic data sets and semantic technologies that model the users' interest using ontologies to link and enrich them with semantic information. We briefly describe the SERUM architecture before we go into detail and explain the semantic technologies used to collect the user behavior and to compute the user interest model.

3.4.2 The SERUM Architecture

The SERUM architecture consists of four building blocks:

- the news crawler,
- the named entity recognition and disambiguation component (NER/NED),
- the user modeling component and
- the semantic recommender.

[16] http://news.google.com/, search conducted on October 19th, 2014

[17] RDFa (or Resource Description Framework - in - attributes) is a W3C Recommendation that allows to embed rich metadata within Web documents.

The news crawler component, provided by Neofonie GmbH[18], collects around 60,000 news articles from German and English news sites every day. *The NER/NED component* [151] identifies and extracts named entities from these news texts and links them to a data set collected from Freebase[19]. Freebase is a semantic encyclopedic data collection, comparable to DBpedia[20]. The data set consists of ≈ 400,000 artists, ≈ 1,700,000 tracks and albums, and ≈ 2,000 genres, connected by ≈ 1,9 million edges. This data is interlinked with the news corpus through the entities detected in news articles using the NER/NED component. The NER/NED associates a Freebase entry to every entity found in an article by linking a Freebase URL to the entity. The news corpus currently contains over 7,200,000 news articles, growing daily by the newly crawled articles, and builds together with the Freebase data the knowledge base for the recommender. The recommendation algorithm itself is explained in detail in Chapter 5 and in [128].

The user modeling component implicitly collects the users' reading behavior to build a user model containing the users' interest in topics or entities. Fig. 3.8 shows the user interface of SERUM with the personalized news stream. Under each news article, all entities are displayed, which are detected in the article. Each user interaction with an article or an entity is tracked and incorporated in the user model. In the current system, we focus on four behavior tracking use cases:

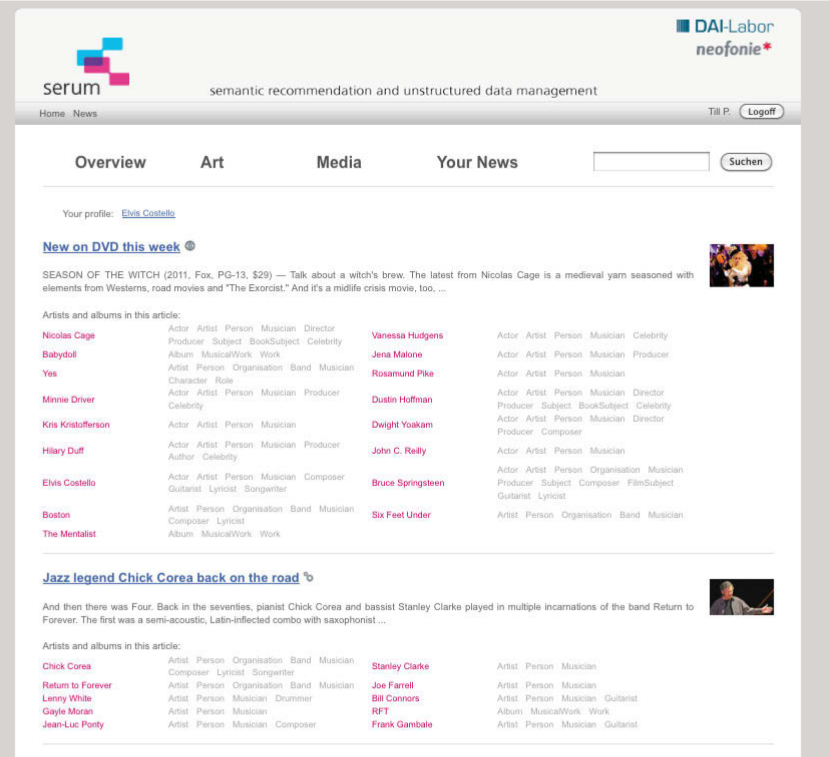
- User clicks on an article: The news and all related entities are marked as interesting.
- User clicks on an article in a list: The clicked article and all related entities are marked as interesting for the user, while all other surrounding articles are marked as less interesting.
- User clicks on recognized entities in an article and
- triggered mouse-over events: Entities clicked by the user or marked by the mouse pointer are given a higher interest rating.

This user feedback is collected using the semantic user behavior tracker described in Section 3.2, which is part of the web application. The data is

[18] http://www.neofonie.de/

[19] http://freebase.com

[20] http://dbpedia.org/



Your profile: Elvis Costello

New on DVD this week

SEASON OF THE WITCH (2011, Fox, PG-13, $29) — Talk about a witch's brew. The latest from Nicolas Cage is a medieval yarn seasoned with elements from Westerns, road movies and "The Exorcist." And it's a midlife crisis movie, too, ...

Artists and albums in this article:

Nicolas Cage	Actor Artist Person Musician Director Producer Subject BookSubject Celebrity	Vanessa Hudgens	Actor Artist Person Musician Celebrity
Babydoll	Album MusicalWork Work	Jena Malone	Actor Artist Person Musician Producer
Yes	Artist Person Organisation Band Musician Character Role	Rosamund Pike	Actor Artist Person Musician
Minnie Driver	Actor Artist Person Musician Producer Celebrity	Dustin Hoffman	Actor Artist Person Musician Director Producer Subject BookSubject Celebrity
Kris Kristofferson	Actor Artist Person Musician	Dwight Yoakam	Actor Artist Person Musician Director Producer Composer
Hilary Duff	Actor Artist Person Musician Producer Author Celebrity	John C. Reilly	Actor Artist Person Musician
Elvis Costello	Actor Artist Person Musician Composer Guitarist Lyricist Songwriter	Bruce Springsteen	Actor Artist Person Organisation Musician Producer Subject Composer FilmSubject Guitarist Lyricist
Boston	Artist Person Organisation Band Musician Composer Lyricist	Six Feet Under	Artist Person Organisation Band Musician
The Mentalist	Album MusicalWork Work		

Jazz legend Chick Corea back on the road

And then there was Four. Back in the seventies, pianist Chick Corea and bassist Stanley Clarke played in multiple incarnations of the band Return to Forever. The first was a semi-acoustic, Latin-inflected combo with saxophonist ...

Artists and albums in this article:

Chick Corea	Artist Person Organisation Band Musician Composer Lyricist Songwriter	Stanley Clarke	Artist Person Musician
Return to Forever	Artist Person Organisation Band Musician	Joe Farrell	Artist Person Musician
Lenny White	Artist Person Musician Drummer	Bill Connors	Artist Person Musician Guitarist
Gayle Moran	Artist Person Musician	RFT	Album MusicalWork Work
Jean-Luc Ponty	Artist Person Musician Composer	Frank Gambale	Artist Person Musician Guitarist

Figure 3.8: SERUM Interface showing recommended news articles and recognized entities.

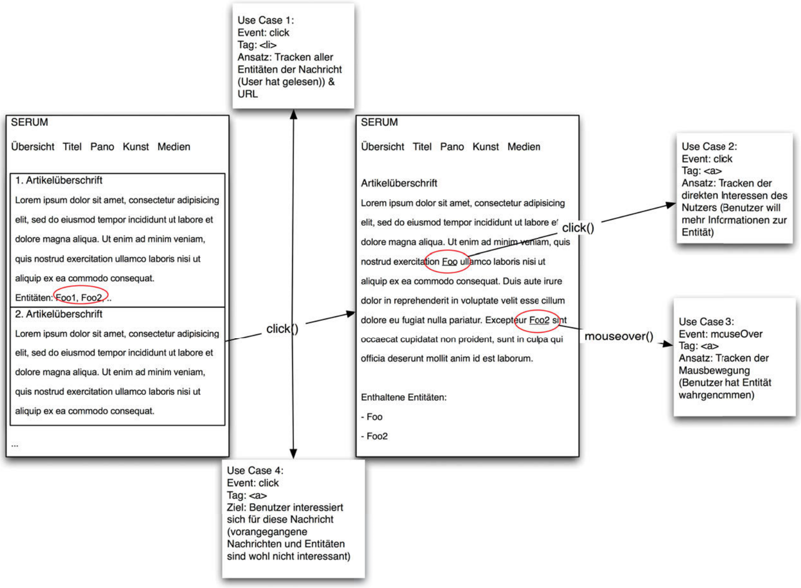

Use Case 1:
Event: click
Tag: <li>
Ansatz: Tracken aller Entitäten der Nachricht (User hat gelesen)) & URL
SERUM
Übersicht Titel Pano Kunst Medien
1. Artikelüberschrift
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Entitäten: Foo1, Foo2, ..
2. Artikelüberschrift
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
...
click()
SERUM
Übersicht Titel Pano Kunst Medien
Artikelüberschrift
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation Foo ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur Foo2 sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Enthaltene Entitäten:
- Foo
- Foo2
click()
Use Case 2:
Event: click
Tag: <a>
Ansatz: Tracken der direkten Interessen des Nutzers (Benutzer will mehr Informationen zur Entität)
mouseover()
Use Case 3:
Event: mouseOver
Tag: <a>
Ansatz: Tracken der Mausbewegung (Benutzer hat Entität wahrgenommen)
Use Case 4:
Event: click
Tag: <a>
Ziel: Benutzer interessiert sich für diese Nachricht (vorangegangene Nachrichten und Entitäten sind wohl nicht interessant)

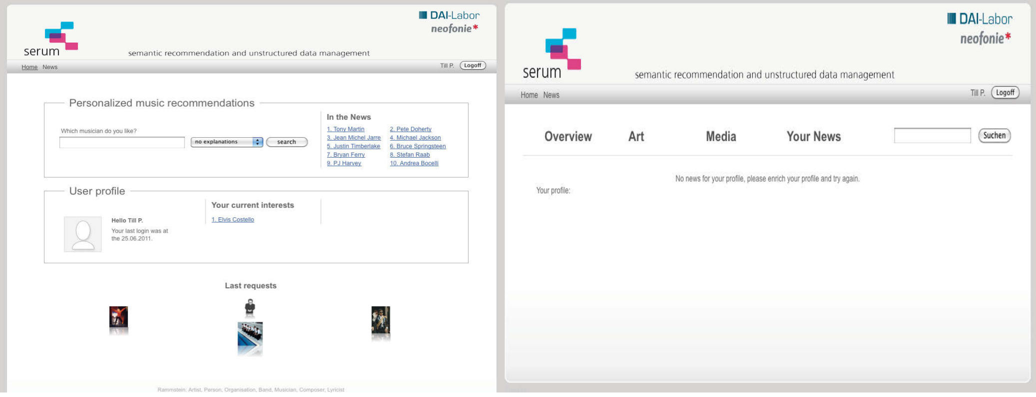
Figure 3.9: Visualization of the SERUM User Tracking use cases.

stored on the server-side in an RDF store using the User Behavior Ontology (UBO), described in Fig. 3.16. We build on the idea presented in [185] to use a distinct behavior model but use a more comprehensive model to not only track events but also to track semantic relations between entities on a webpage as presented in [165]. UBO describes all events relevant for modeling the user behavior such as user clicks or mouse-over events. Events, triggered by the user (e.g. clicks) are linked to news articles and named entities (e.g. artists in the news article) the user interacted with.

Based on a statistical interaction analysis the user behavior events are aggregated to identify named entities (e.g. topics, musicians and genres) the user is interested in. The analysis includes the last n sessions of the user (in our current system n is set to 5) where the interaction of a user is analyzed and the entities are ranked according to the interaction frequency. The analysis also includes a time aspect where an interaction has a higher weight if the session is a current one. Furthermore, we deploy semantic data (from Freebase) to extend the knowledge about identified named entities to produce a richer user model. Thus, musicians recognized to be interesting to the user are expanded with data about produced albums and collaborating artists. For example, if the user only stated interest in "Madonna", we can add genre information (e.g. pop) and information about collaborations with other artists. These enriched user profiles are used as the input for our graph-based recommender. The more information in the user profile, the more likely it is to find related news for a user. The news recommendation strategy is based on the recentness of the news as well as the correlation of computed interests and their occurrence in the news. Based on the defined architecture, we introduce the use cases that we showcase with Serum in the next section.

3.4.3 Serum Use Case - User Behavior Collection

In order to explain the interaction of the semantic tracking component with the news recommendation system, we walk through the first and fourth use case and detail the tracked user interaction, the resulting user model and the recommendations. As mentioned in Section 3.4.2, Serum is a personalized news recommender where the user profile is created by tracking and analyzing user behavior. Initially, after the first login, the user profile and the personalized news stream is empty as depicted in Fig. 3.10. The picture shows the empty user profile on the left and the empty personalized news stream. To create the user profile, the user has to interact with Serum, to read news or to search for artists.



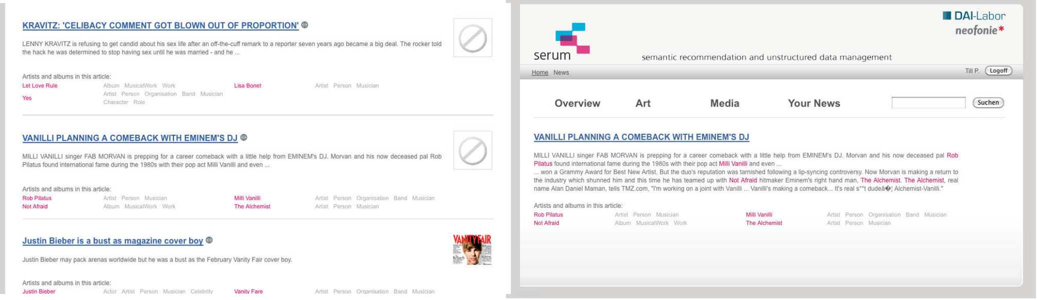
Your last login was at the 25.06.2011.

Rammstein: Artist, Person, Organisation, Band, Musician, Composer, Lyricist

No news for your profile, please enrich your profile and try again.

Figure 3.10: Serum Personalization News: After the first login, no news are recommended (right side) because no profile exists (left side) .

If the user starts reading, their first interaction is with a list of news articles where they can choose what to read. The Serum news list shows the article, an abstract and a list of entities (artists) that are found within the text. Fig. 3.11 shows the news list overview on the left hand side and a detailed view of the selected article on the right hand side.



KRAVITZ: 'CELIBACY COMMENT GOT BLOWN OUT OF PROPORTION'

LENNY KRAVITZ is refusing to get candid about his sex life after an off-the-cuff remark to a reporter seven years ago became a big deal. The rocker told the hack he was determined to stop having sex until he was married - and he ...

Artists and albums in this article:

Let Love Rule	Album MusicalWork Work	Lisa Bonet	Artist Person Musician
Yes	Artist Person Organisation Band Musician Character Role		

VANILLI PLANNING A COMEBACK WITH EMINEM'S DJ

MILLI VANILLI singer FAB MORVAN is prepping for a career comeback with a little help from EMINEM's DJ. Morvan and his now deceased pal Rob Pilatus found international fame during the 1980s with their pop act Milli Vanilli and even ...

Artists and albums in this article:

Rob Pilatus	Artist Person Musician	Milli Vanilli	Artist Person Organisation Band Musician
Not Afraid	Album MusicalWork Work	The Alchemist	Artist Person Musician

Justin Bieber is a bust as magazine cover boy

Justin Bieber may pack arenas worldwide but he was a bust as the February Vanity Fair cover boy.

Artists and albums in this article:

Justin Bieber	Actor Artist Person Musician Celebrity	Vanity Fare	Artist Person Organisation Band Musician

DAI-Labor
neofonie*
serum
semantic recommendation and unstructured data management
Home News
Till P. Logoff
Overview Art Media Your News
Suchen

VANILLI PLANNING A COMEBACK WITH EMINEM'S DJ

MILLI VANILLI singer FAB MORVAN is prepping for a career comeback with a little help from EMINEM's DJ. Morvan and his now deceased pal Rob Pilatus found international fame during the 1980s with their pop act Milli Vanilli and even ...
... won a Grammy Award for Best New Artist. But the duo's reputation was tarnished following a lip-syncing controversy. Now Morvan is making a return to the industry which shunned him and this time he has teamed up with Not Afraid hitmaker Eminem's right hand man, The Alchemist. The Alchemist, real name Alan Daniel Maman, tells TMZ.com, "I'm working on a joint with Vanilli ... Vanilli's making a comeback... It's real s**t dudeâ�¦ Alchemist-Vanilli."

Artists and albums in this article:

Rob Pilatus	Artist Person Musician	Milli Vanilli	Artist Person Organisation Band Musician
Not Afraid	Album MusicalWork Work	The Alchemist	Artist Person Musician

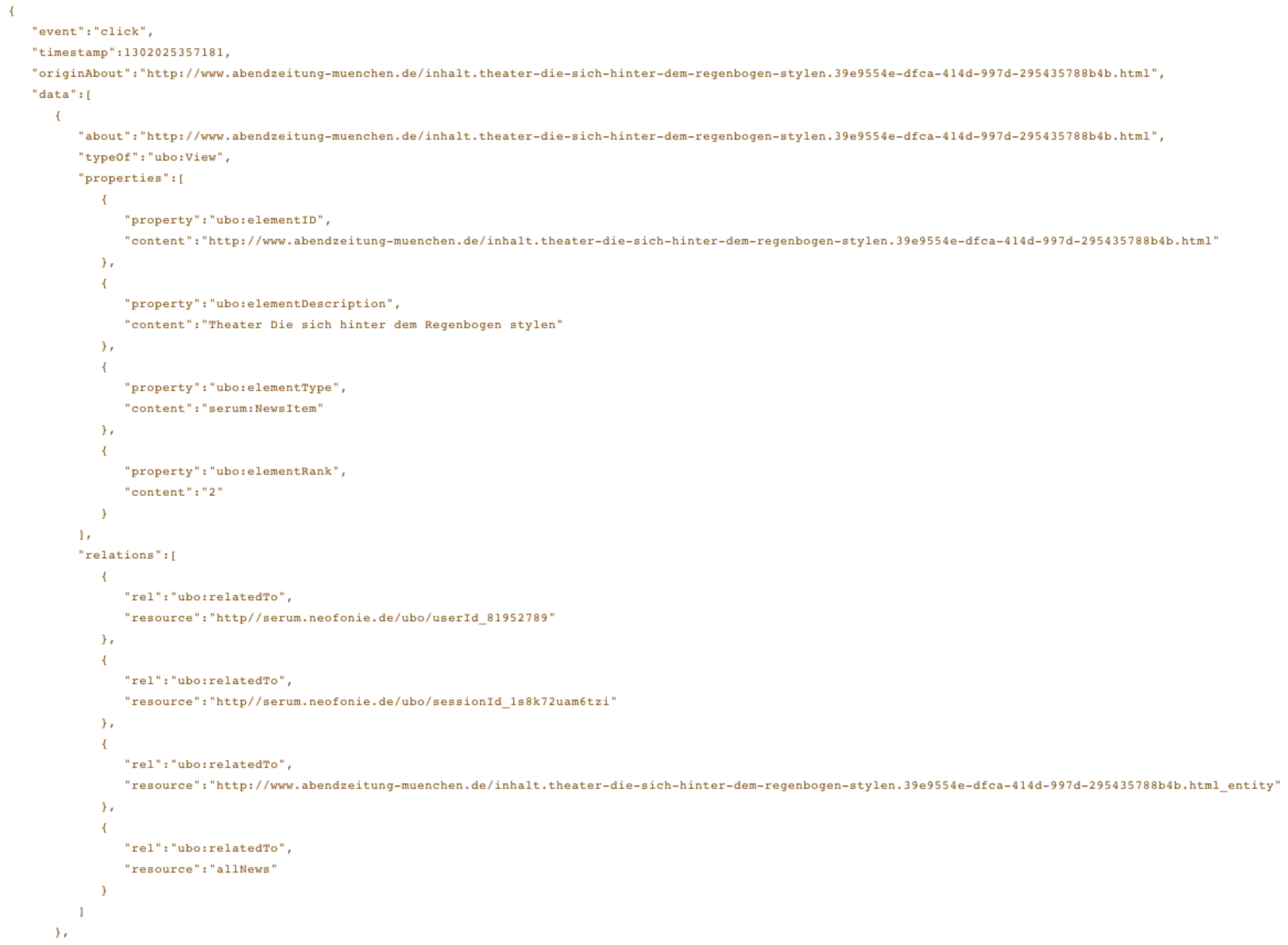
Figure 3.11: Serum News List and article with the artists that are part of the article.

When the user clicks on an article, that article, the position of the article and the surrounding articles are tracked and send back to the server. Fig. 3.12 shows an example JSON snippet that is send back to the server for profile creation. This tracked information allows to start building a user profile as the read article, and the connected artists, are getting a positive weight. The articles, and connected entities, surrounding the read article getting a negative weight, as they were in the users viewport but were not as interesting as the read article.

Apart from the tracked article information, information about the user and the used device is also tracked and sent back to the server to assign the data to one user (see Fig. 3.13). While users are reading the article, SERUM also tracks the mouse movement and if they hover over an artist. This is also sent back to the system as it may indicate that this artist is of special interest [59]. A direct click on an artist, which leads to an extra info site about the artist, is also tracked and treated which much higher weight for the user profile creation.

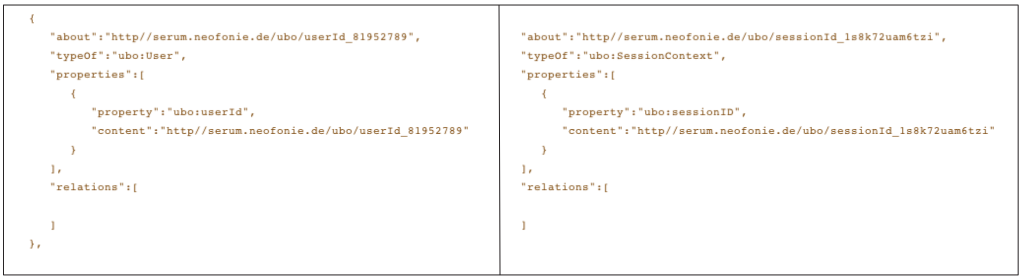
This information, tracked by our tracking system builds the base for the creation of a user interest profile. The used profile creation mechanism follows the presented use cases, e.g., clicks on an article mark all artists as interesting for a user while artists from article surrounding the clicked one are marked as less interesting. The resulting user profile is shown in Fig. 3.14.

The created profile is used to create the personalized news stream, shown in Fig. 3.15. The news is based on the user profile, which is a weighted



```
{
   "event":"click",
   "timestamp":1302025357181,
   "originAbout":"http://www.abendzeitung-muenchen.de/inhalt.theater-die-sich-hinter-dem-regenbogen-stylen.39e9554e-dfca-414d-997d-295435788b4b.html",
   "data":[
      {
         "about":"http://www.abendzeitung-muenchen.de/inhalt.theater-die-sich-hinter-dem-regenbogen-stylen.39e9554e-dfca-414d-997d-295435788b4b.html",
         "typeOf":"ubo:View",
         "properties":[
            {
               "property":"ubo:elementID",
               "content":"http://www.abendzeitung-muenchen.de/inhalt.theater-die-sich-hinter-dem-regenbogen-stylen.39e9554e-dfca-414d-997d-295435788b4b.html"
            },
            {
               "property":"ubo:elementDescription",
               "content":"Theater Die sich hinter dem Regenbogen stylen"
            },
            {
               "property":"ubo:elementType",
               "content":"serum:NewsItem"
            },
            {
               "property":"ubo:elementRank",
               "content":"2"
            }
         ],
         "relations":[
            {
               "rel":"ubo:relatedTo",
               "resource":"http//serum.neofonie.de/ubo/userId_81952789"
            },
            {
               "rel":"ubo:relatedTo",
               "resource":"http//serum.neofonie.de/ubo/sessionId_1s8k72uam6tzi"
            },
            {
               "rel":"ubo:relatedTo",
               "resource":"http://www.abendzeitung-muenchen.de/inhalt.theater-die-sich-hinter-dem-regenbogen-stylen.39e9554e-dfca-414d-997d-295435788b4b.html_entity"
            },
            {
               "rel":"ubo:relatedTo",
               "resource":"allNews"
            }
         ]
      },
```

Figure 3.12: JSON of the SERUM User Tracking uses cases.



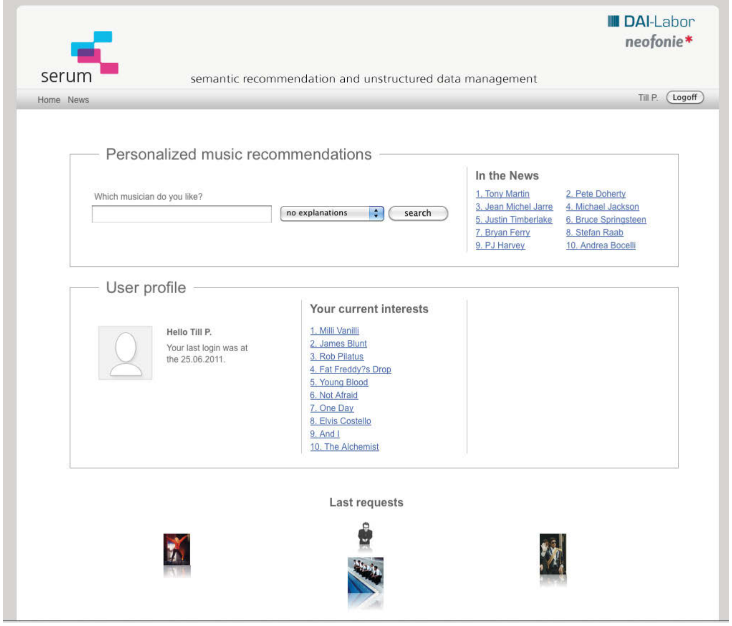
```
{
   "about":"http//serum.neofonie.de/ubo/userId_81952789",
   "typeOf":"ubo:User",
   "properties":[
      {
         "property":"ubo:userId",
         "content":"http//serum.neofonie.de/ubo/userId_81952789"
      }
   ],
   "relations":[

   ]
},
```

```
"about":"http//serum.neofonie.de/ubo/sessionId_1s8k72uam6tzi",
"typeOf":"ubo:SessionContext",
"properties":[
   {
      "property":"ubo:sessionID",
      "content":"http//serum.neofonie.de/ubo/sessionId_1s8k72uam6tzi"
   }
],
"relations":[

]
```

Figure 3.13: JSON of the SERUM User Tracking uses cases. User and Session Information



Personalized music recommendations

Which musician do you like?

In the News

1. Tony Martin	2. Pete Doherty
3. Jean Michel Jarre	4. Michael Jackson
5. Justin Timberlake	6. Bruce Springsteen
7. Bryan Ferry	8. Stefan Raab
9. PJ Harvey	10. Andrea Bocelli

User profile

Hello Till P.

Your last login was at the 25.06.2011.

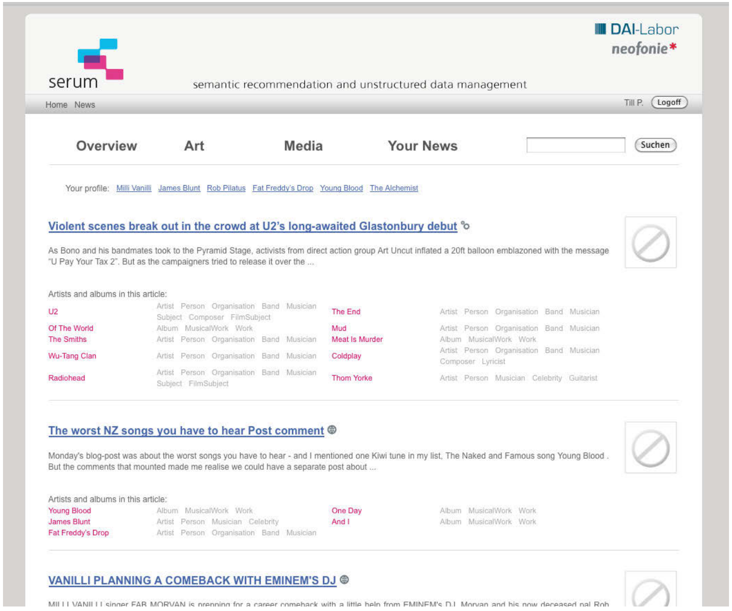
Your current interests

1. Milli Vanilli
2. James Blunt
3. Rob Pilatus
4. Fat Freddy?s Drop
5. Young Blood
6. Not Afraid
7. One Day
8. Elvis Costello
9. And I
10. The Alchemist

Last requests

Figure 3.14: Serum: The user profile after reading some articles.

profile and the in chapter 5 presented graph-based algorithm to enrich user profiles.



Your profile: Milli Vanilli James Blunt Rob Pilatus Fat Freddy's Drop Young Blood The Alchemist

Violent scenes break out in the crowd at U2's long-awaited Glastonbury debut

As Bono and his bandmates took to the Pyramid Stage, activists from direct action group Art Uncut inflated a 20ft balloon emblazoned with the message "U Pay Your Tax 2". But as the campaigners tried to release it over the ...

Artists and albums in this article:

U2	Artist Person Organisation Band Musician Subject Composer FilmSubject	The End	Artist Person Organisation Band Musician
Of The World	Album MusicalWork Work	Mud	Artist Person Organisation Band Musician
The Smiths	Artist Person Organisation Band Musician	Meat Is Murder	Album MusicalWork Work
Wu-Tang Clan	Artist Person Organisation Band Musician	Coldplay	Artist Person Organisation Band Musician Composer Lyricist
Radiohead	Artist Person Organisation Band Musician Subject FilmSubject	Thom Yorke	Artist Person Musician Celebrity Guitarist

The worst NZ songs you have to hear Post comment

Monday's blog-post was about the worst songs you have to hear - and I mentioned one Kiwi tune in my list, The Naked and Famous song Young Blood . But the comments that mounted made me realise we could have a separate post about ...

Artists and albums in this article:

Young Blood	Album MusicalWork Work	One Day	Album MusicalWork Work
James Blunt	Artist Person Musician Celebrity	And I	Album MusicalWork Work
Fat Freddy's Drop	Artist Person Organisation Band Musician		

VANILLI PLANNING A COMEBACK WITH EMINEM'S DJ

MILLI VANILLI singer FAB MORVAN is prepping for a career comeback with a little help from EMINEM's DJ. Morvan and his now deceased pal Rob

Figure 3.15: Serum: Personalized news.

3.4.4 Conclusion

The SERUM project shows that with the combination of our tracking system and the UBO, the creation of user interests profiles become simply and effective. With no visible intervention on the website, detailed tracking of user actions is possible. This is the main requirement of our tracking system. Of course, below the surface the website structure has to be extended with semantic information using microformats or RDFa. But, relying on the semantic tracking solution, with only a few read articles, the user profile already reflects general interests of the user and allows us to offer a personalized news stream filtering the huge amount of articles. A first non-representative test during the project showed that the created profile satisfies expectation and the recommended news match the profiles. While the presented scenario in Section 3.4.3 only showed the tracking of mouse events, the SERUM system also tracks searches for artists and uses this information also for the profile creation. As a search is an explicit action, the artists the user searches for are get an higher weight in the user profile. This complex tracking is unobtrusive and transparent for the user, which was one requirement of our
]

tracking solution. The management of the tracked information using the UBO allows the usage of this data for future personalization in different applications. If a user registers for a new application, his previously collected behavior data can be used to adapt the UI to personal preferences or to compute recommendations.

3.5 Closing Discussion

In this chapter, we presented a course of action to extend the Web Usage Mining (WUM) process to a Semantic Web Usage Mining (SWUM) process. We presented a system to not only track user behavior but also track meaning behind the actions. Our RDFa based approach tracks not only the actual actions but also connected elements describing the user's intention. This tracked data is stored in a novel ontology. The User Behavior Ontology (UBO) allows to store the tracking data and the meaning. Based on the news recommendation prototype SERUM, we showed that the extension to SWUM helps generating recommendations and creating user profiles with less need for explicit user interaction. With the usage of semantic information on both ends of the SERUM system, RDFa on the client-side and the UBO and semantic encyclopedic data in the back-end, we are able to build richer profiles, which help to improve the recommendation quality and user satisfaction. The use of semantic encyclopedic data allows us to extend our knowledge about a user. The collection of user behavior using RDFa allows tracking not only information the user directly interacted with but also information that is related to an interaction. Thus, we have more information about the user and can compute precise interests.

While in this chapter we discussed how to get more information about user interests based on user behavior, in the next chapter we focus on the user model itself. We will discuss two questions: 'How can we aggregate information from different applications?' and 'How can we build a user model for the social web?'.

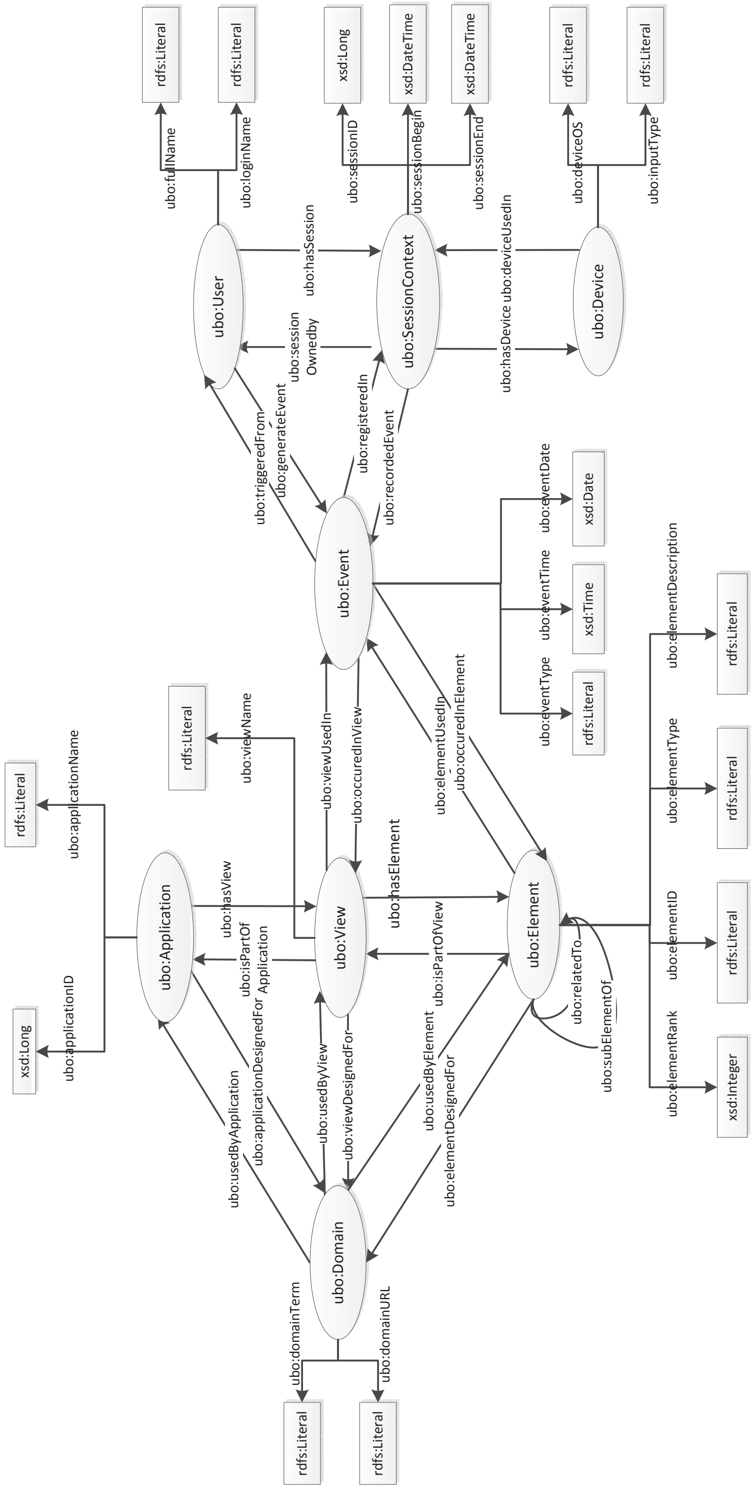


Figure 3.16: UBO: User Behavior Ontology – An ontology for user behavior collection

Chapter 4

User Models for the Social Semantic Web

With the growing impact of the Social Web, or Web 2.0, on our every day life, people start to use more and more different web based services like Facebook[1], Twitter[2], Flickr[3] or blogs. They use these services to express their opinion, communicate with others and share pictures with friends. Thereby, they generate and distribute personal and social information like interests, social contacts, preferences and personal goals [3]. This user information is usually stored in a user profile deeply buried within every service, only accessible through a service's User Interface (UI) or API. This affects the user's ability to keep track of their personal information. They loose the overview of what information is stored where and what is public and what private, which leads to open privacy and security challenges. Users who have no overview of the data stored cannot control what data is publicly available and thus, information can be shared by accident. However, the personal information distributed over different services represents an untapped store of knowledge that could be used to enhance personalization and recommendation for existing services.

In this chapter, we focus on the second layer of the adaptive system architecture, illustrated in Fig. 4.1, the *Data Representation* layer. We want to answer the question how can we make use of the personal information a

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

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

[3] http://www.flickr.com

single user is spreading all over the Social Web every day. We investigate what is needed from a user model point of view to support user data sharing and aggregation to enhance personalization and recommendation services. In this chapter we give an introduction to current approaches to the problem. In Section 4.1, we introduce the problem of user model aggregation and standardization as a motivation for the work presented in Section 4.2 and Section 4.3. In Section 4.2, we present a meta-model that allows to aggregate information from different applications without the need for those applications to use on model. Also a system using this model to aggregate and utilize the user data is presented. In Section 4.3, we present a study of 17 social applications to define requirements and attributes for a common user model that allows sharing of user data and analyze what is needed to enhance user model aggregation approaches. As a result, we present a comprehensive user model especially fitted to the needs of the Social Web. Furthermore, we present a specialized WordNet[4] for the user modeling domain as part of the user model to support user model aggregation.

The main contributions for this chapter have been published in [64, 163, 166, 58, 158].

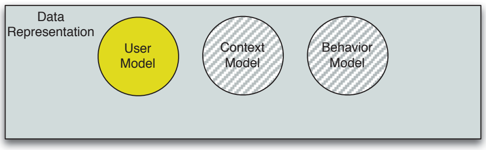


Figure 4.1: Parts of the adaptive system that are discussed in Chapter 4.

4.1 Approaches to User Model Aggregation and Standardization

Until the turn of the millennium, most personalization and recommendation research focused on user information available in one application and how to use this information to enhance personalization and recommendation quality. With the advent of the Social Web, or Web 2.0, user information became highly distributed over several applications and research started to explore

[4] WordNet is a lexical database containing information about words and how they are related, http://wordnet.princeton.edu/

cross-system personalization approaches using combined data from different applications. As this user information is typically stored in proprietary formats defined by each application it needs to be aggregated to get a holistic view on the data, we need mechanisms to aggregate different user profiles. These aggregated user profiles have to be presented in a unified way to have an inter-application understanding of the stored information [152]. Such aggregated user profiles are also the basis for personalization of applications and recommendations [160]. In the research fields of user modeling and user model aggregation, different approaches have been proposed to address the problem of user model heterogeneity and aggregation. These approaches can be categorized into two types [121]:

- **Standardization of user models**: A common and shared user model standard for all applications.
- **User model mediation**: A set of techniques to transform or convert data from one user model to another format. This is a practical approach to solve the problem of heterogeneity and allows the aggregation of different models.

The work presented in this chapter is in alignment with the first strategy, the standardization approach. The second strategy, the mediation approach mainly driven by Berkovsky et al. [27, 28] is a more practical approach. It aims to build a integrated user model suitable for a specific goal, e.g. recommendation of music. This integrated user model is based on data collected from different applications and aggregated by specialized software components. These software components transform data from one representation into a target representation. This approach solves the heterogeneity problem by having specialized software components for each transformation. The shortcoming is that for each data field to be transformed a separate component has to be developed which can lead to immense computational efforts. The goal of this work is to research and develop more general approaches to the user model aggregation problem. We therefore focus on common, standardized models that are suited for the Social and Semantic Web. In the remainder of this section, we give an introduction into the field of standardized approaches.

The standardized approach can again be subdivided into two aggregation strategies. The first strategy proposes the use of standardized user models that all involved applications must agree on. The second strategy deals with the mediation of different user model representations using meta-models that connect user data from one application with data from another application, in the same domain, or across domains. The standardization approach has a long research history starting with early works of simple user modeling shells [77] to more sophisticated user modeling servers [107, 78]. The standardization approach involves no computational effort to aggregate data as all data already is in the same format. An effort in this direction is the General User Modeling Ontology (GUMO) created by Heckman et al. [89, 90, 91]. GUMO is a comprehensive user model that intends to cover all aspects of a user's life. The user dimensions covered range from contact information and demographics over abilities, personality right up to special information like mood, nutrition or facial expressions. GUMO is at the time of this writing the most comprehensive generic user modeling ontology. Another approach that came up with the Web 2.0 is the Friend-of-a-Friend (FOAF) ontology. FOAF is a lightweight model that is integrated on the website, the application's user interface, using RDFa. FOAF covers basic user information like contact information, basic demographics and allows to specify some social relations like group membership or 'knows' relations to other FOAF profiles. GUMO, which represents the most generic user model, covers only some parts of information that are needed for the Social Web. Especially the *Interest* dimension (in music, books, etc.) and user information like accounts for different Social Web applications, which are crucial, as we show in Section 4.3.1, are completely missing. FOAF, which is designed for a Web use, is too simplified. FOAF has a 'knows' relationship, which defines a social relation, but the type of the relation remains unclear. Also no user needs and goals can be defined, which is part of many social applications as we will see in Section 4.3.1. The Unified User Context Model (UUCM) [137, 143], introduced by Niderée et al., is a centralized and extensible multi-dimensional user model for aggregating the partial user models collected by individual personalization systems. The UUCM defines two levels: the abstract and the concrete level. The abstract level defines the principal elements of the UUCM that are: user context, user model features, main characteristics for feature representation, and user model dimensions. In order to be used for cross-system personalization, this level specifies a shared ontology and all systems depend on this model. The concrete level defines a group of UUCM dimensions and features that include not only users' interests, but also tasks and relations to other entities and relevant user communities. Different features are modeled with the use of name/value pair. Each personalization system has to build upon its user model the UUCM structure to be able to share data via a Cross-System Communication Protocol (CSCP).

The second strategy is to build meta-models that allow defining how application-dependent user data corresponds to user data from another application. This has the advantage that applications are not forced to adopt a predefined generic user model and can rely on their own model. In Section 4.2, we present an aggregation ontology which gives applications the possibility to define a model, which describes how information in different profiles is related and how data can be aggregated. Furthermore, the ontology not only allows to define relations between data in different application models but also to define the overlap, the similarity, of the modeled information. So it is possible to define that the field 'interests' in one application and the field 'music interests' in another, is related but only to a certain degree as 'music interests' is only subset of 'interests'. In [206], van der Sluijs et al. present the Generic User model Component (GUC) which builds a central component where all applications have to subscribe to and describe their user model via a schema defining the data structure of the user models for different applications. The authors also suggest the possibility to use different matching and merging techniques to map input schemas and create a merged schema as the union of the input schemata and to construct combined ontologies of the application schemata. While the meta-model approach seems to be a more practical one to achieve a semantic and syntactic interoperability, the big disadvantage is that is needs a lot of effort to connect all the different user models. This work currently has to be done manually or semi-manually and must be repeated for every new application user model.

To summarize: both strategies, common model and meta-model, have shortcomings. Because of big differences, regarding the covered user information and representation forms in different applications, the development of a commonly accepted ontology, covering all aspects of user modeling for all domains is not feasible as the adaption rate of GUMO shows. A meta-model approach, without automatic aggregation mechanisms, is solely applicable in small settings where only few applications are connected and not for the Social Web. We propose a middle way: We need a new "common" user model that combines aspects of the presented approaches and focuses on a special domain, the Social Web. Also, by focusing on a special domain, we want support for automatic, or at least a semi-automatic user model aggregation by defining a structure that allows finding relations between different user model concepts.

In the next section we first present a new meta-model that allows to describe in great detail how user models from different applications are connected and how big the overlap of information is. For example one user model has a field 'name' and the other model a field 'last_name', then it is not a one hundred percent match as the first field could contain first and last name. In Section 4.3 we present a new generic user model ontology for the Social Web and a User Model WordNet that helps to map information automatically.

4.2 User Model Aggregation

In this section, we present a user-centric, thus privacy preserving, system that consists of a semantic layer to aggregate user models and a personal user interface to visualize the profile information. The semantic layer aggregates user models from different web applications to allow access to information stored in different user profiles in a unified way. It therefore builds on a new ontology that allows connecting user models from different applications and enables a unified access. We also present a UI utilizing the ontology and the connected profiles allowing users to keep track of personal information stored in different applications. This helps users to control their personal data and thus it helps to prevent unintended data sharing. The section is structured as follows: We first describe in detail our semantic approach to aggregate user models and how to access this information in a private and secure way. Then, we show a system that uses the aggregated information to give users an overview about the personal information stored in different applications.

4.2.1 Semantic User-Centric Data Management

To aggregate information, we developed an ontology, see Section 4.2.1 that serves as the basis for our system. For the sharing of information, we build a privacy-preserving system, presented in Section 4.2.2, which manages and visualizes data and shares information between applications only with user consent. Fig. 4.2 shows three user profiles with three attributes each, containing personal information and interest information which can be aggregated utilizing our approach. In the following section, we present our ontology and describe how the aggregation process works. At the end of this section, the three models in Fig. 4.2 are connected.

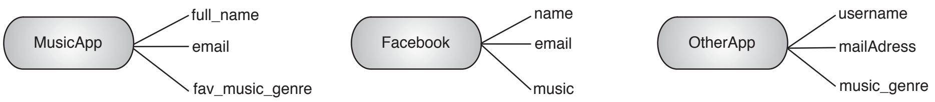


Figure 4.2: Three different user profiles containing personal information and music interests.

User model aggregation: The Profile Data Model

We first present our self-developed generic ontology, the Profile Data Model (PDM) that gives us the possibility to define a model which describes how information in different profiles is related and how data can be aggregated. Fig. 4.3 visualizes our ontology, the PDM. The PDM not only allows us to define relations between models but also to define a degree of similarity between information and to determine the information source (the application that provided the data). This aggregation model with descriptions about coherences between the user profile data is the basis for a system presented in Section 4.2.2 which allows to show users their personal information stored in different applications and to manage data access and sharing.

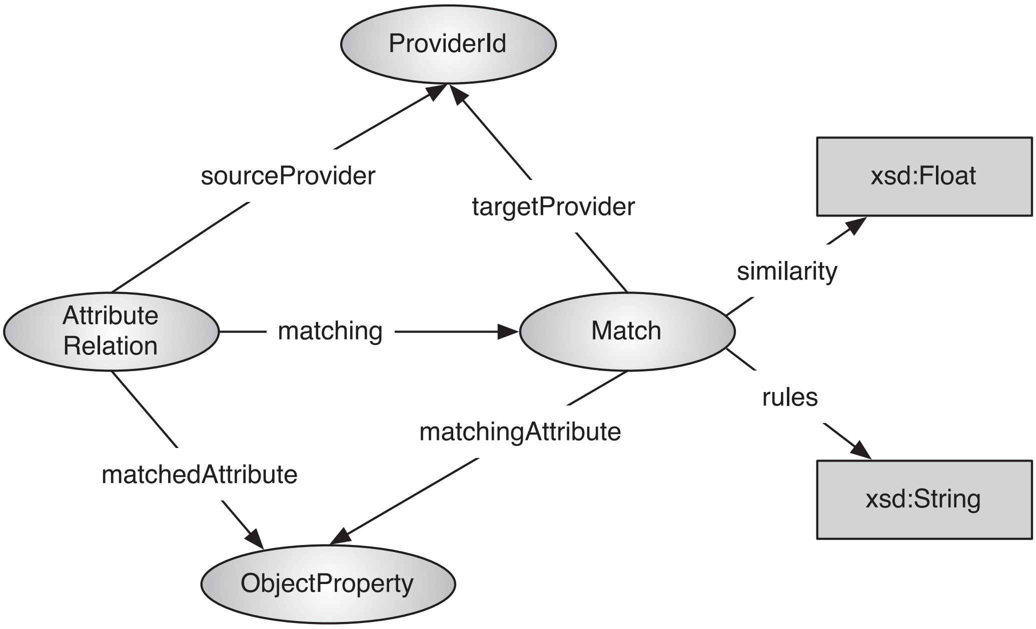


Figure 4.3: Ontology for Profile Aggregation.

An important extension to existing ontological approaches, e.g. [206], is the entity **Match**, which allows to add extra knowledge to the model. Extra knowledge can be a similarity measure of the related data in two different user profiles or instructions. Instructions can be a set of predefined rules [123] describing how to aggregate information. Such extra information is an important information for later access and handling of the information. If for instance an application ask for the favorite music genres of a user, our aggregation model could return the fields 'fav_music_genre' and 'music_genre' and indicate that the field 'music' may contain also interesting information (example is based on Fig. 4.2). The requesting application can then decide if it wants to access 'music' or not. The field 'music' could contain only favorite artists of a user, in such a case the requesting application can't use the information or needs extra effort to get genre information, e.g. from an external Semantic Web resource such as Freebase. Table 4.1 gives a detailed description about all entities and relations of the ontology.

	Location Data Properties	
Predicate	Object Type	Description
ProviderId	rdfs:Literal	The description of an UM provider with name and id. For example Facebook.
AttributeRelation	rdfs:Literal	Defines a relation between the requested attribute (matchedAttribute) of a provider (sourceProvider) with two or more attributes. Connected attributes can be from different providers or only from one provider.
Match	rdfs:Literal	Encapsulates different attribute relations with extra information like similarity, or rules how to aggregate data.
owl:ObjectProperty	rdfs:Literal	Defines the relations between instances of two classes.
sourceProvider	rdfs:Literal	The application identifier of the application hosting the UM.
targetProvider	rdfs:Literal	UM provider of the attribute similar to an requested attribute.
matching	rdfs:Literal	Relation between an AttributeRelation Entity and a Match Entity.
similarity/rules	rdfs:Literal	The similarity attribute defines the degree of similarity between two attributes. The rules attribute defines aggregation rules for the profiles

matchingAttribut	rdfs:Literal	Defines the special application attribute that corresponds to the aggregation model attribute.
matchedAttribut	rdfs:Literal	Defines the attribute that can be requested by other applications to get similar attributes from different applications.

Table 4.1: Entities and relations of the PDM ontology

User profile aggregation with the PDM ontology

Based on this ontology, we can define a concrete model that allows us to aggregate user profiles from different applications and access the information in a unified way. To outline the approach, we exemplarily connect the three user profiles shown in Fig. 4.2 (MusicApp, Facebook, OtherApp) containing personal (name, mail) and interest (music) information.

The actual definition of the model is a straight-forward process. First, one has to analyze the given structure of the different user profiles that should be connected. The goal is to find similar attributes in different profiles that contain similar data. For attributes where the contained information is only partly related, a similarity measure has to be defined. The similarity measure is a substantial information for the data management and visualization process as it is an important indicator for the system on how to handle the data. Such a similarity definition can be done manually, semi-automatically or automatically [18]. The aggregation of the profiles can be automated to some extent [145]. In this scenario, we perform the aggregation manually. We have two information blocks, personal information and music interests, that can be aggregated. To aggregate the music information, we define a new **AttributeRelation** called ’music_favorite_genres’ in our aggregation model (AM). We define **matchedAttribute** (AM#music_favorite_genres) and **sourceProvider** (AM#AM_ID) entries accordingly, which are needed to access the model and retrieve information. Within the encompassing AttributeRelation ’music_favorite_genres’ we define the matching attributes from the different profiles. The attributes are added with **Match** entities

```
<pdm:AttributeRelation rdf:ID="music_favorite_genres">
   <pdm:matchedAttribute rdf:resource="AM#music_favorite_genres"/>
   <pdm:sourceProvider rdf:resource="AM#AM_ID"/>
      <pdm:matching>
         <pdm:Match>
           <pdm:similarity>0.9</pdm:similarity>
           <pdm:targetProvider rdf:resource="musicApp#providerId"/>
           <pdm:matchingAttribute rdf:resource="musicApp#fav_music_genres"/>
         </pdm:Match>
        <pdm:Match>
           <pdm:similarity>0.3</pdm:similarity>
           <pdm:targetProvider rdf:resource="facebook#providerId"/>
           <pdm:matchingAttribute rdf:resource="facebook#music"/>
     </pdm:Match>
     <pdm:Match>
           <pdm:similarity>0.6</pdm:similarity>
           <pdm:targetProvider rdf:resource="otherApp#providerId"/>
           <pdm:matchingAttribute rdf:resource="otherApp#music_genres"/>
     </pdm:Match>
 </pdm:matching>
</pdm:AttributeRelation>
```

Figure 4.4: Aggregation of music interests from 3 different profiles using the PDM ontology.

and have a similarity value defined. The similarity values differ from the ones shown in Fig. 4.2 as they are describing the similarity to the newly created AttributeRelation.

Fig. 4.4 shows the resulting model that describes the relations (Fig. 4.5) of the music interest attributes between the profiles. The aggregation of the personal information attributes follows this process. Once the model is defined, it is integrated into a system that offers a web-service API to access the information.

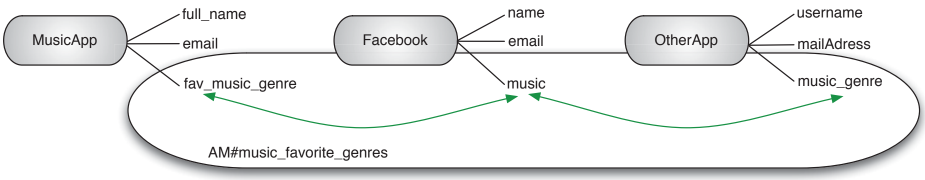


Figure 4.5: The different music interests in the user profiles aggregated using the ontology.

4.2.2 My Personal User Interface - Show Case for Model Aggregation

"My Personal User Interface" is a system that uses the presented PDM ontology, to visualize personal user information distributed over different applications. "My Personal User Interface" has the goal to assist users to

- keep track of applications they have,
- stay in control over their personal data,
- control the information flow of personal data.

The system architecture is depicted in Fig. 4.6. It consists of two main blocks, the Profile Exchange (PE) and the Profile Management (PM) component. The PE is responsible for the secure and controlled data sharing and the PM for the aggregation and visualization of profile information. Therefore, to support people to have an overview over their applications and data "My Personal User Interface" connects data from different applications using the PDM ontology. But not only the aggregation but also the visualization of the data is important as only with an easy to understand UI, a user is able to stay in control of his data. Therefore, "My Personal User Interface" offers different views on the data. The main UI is split into a top and bottom view, see Fig. 4.7.

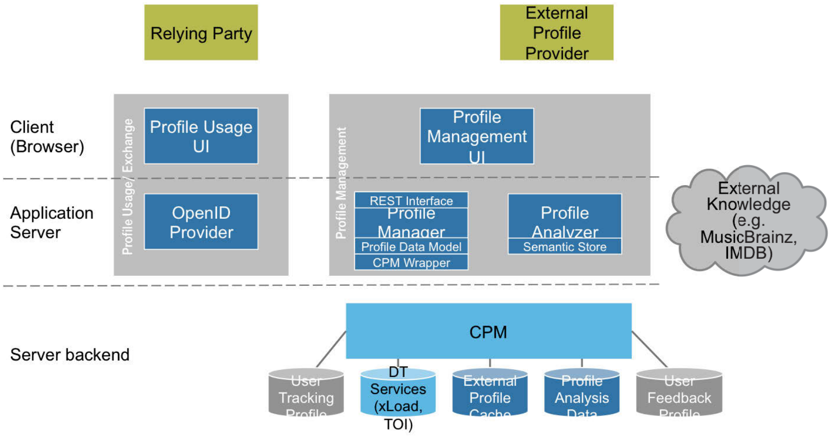
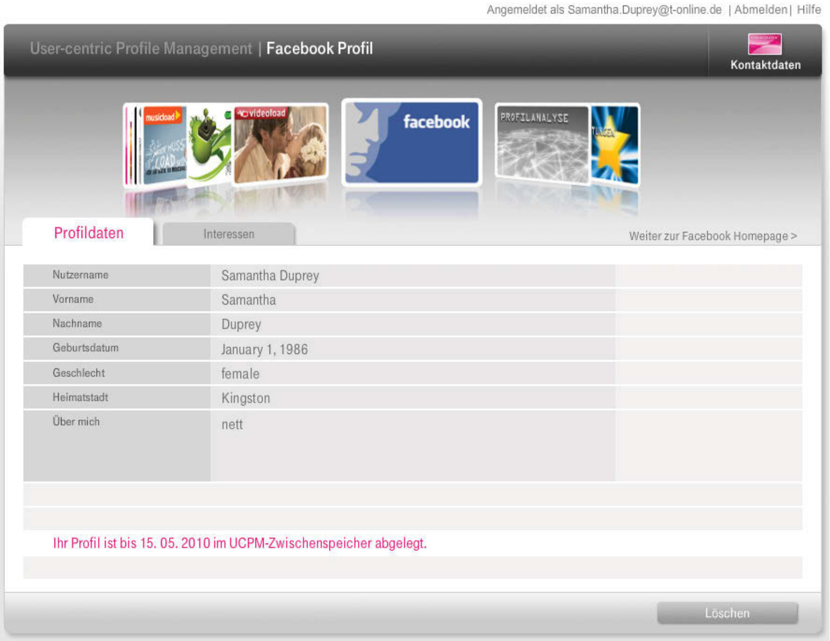


Figure 4.6: Architecture of the My Personal User Interface System.

The top view is a coverflow element showing the different applications of a user. The coverflow allows selecting an application and getting an overview of the personal information stored in it. This personal information is presented in the bottom view of the UI. For example, Fig. 4.7 shows personal contact information stored in Facebook. We adopted the information card metaphor [108] to visualize the different applications of the user and to visualize the stored personal information.


Angemeldet als Samantha.Duprey@t-online.de | Abmelden| Hilfe
User-centric Profile Management | Facebook Profil
Kontaktdaten
musicload
videoload
facebook
PROFILANALYSE
Profildaten
Interessen
Weiter zur Facebook Homepage >
Löschen

Nutzername	Samantha Duprey
Vorname	Samantha
Nachname	Duprey
Geburtsdatum	January 1, 1986
Geschlecht	female
Heimatstadt	Kingston
Über mich	nett

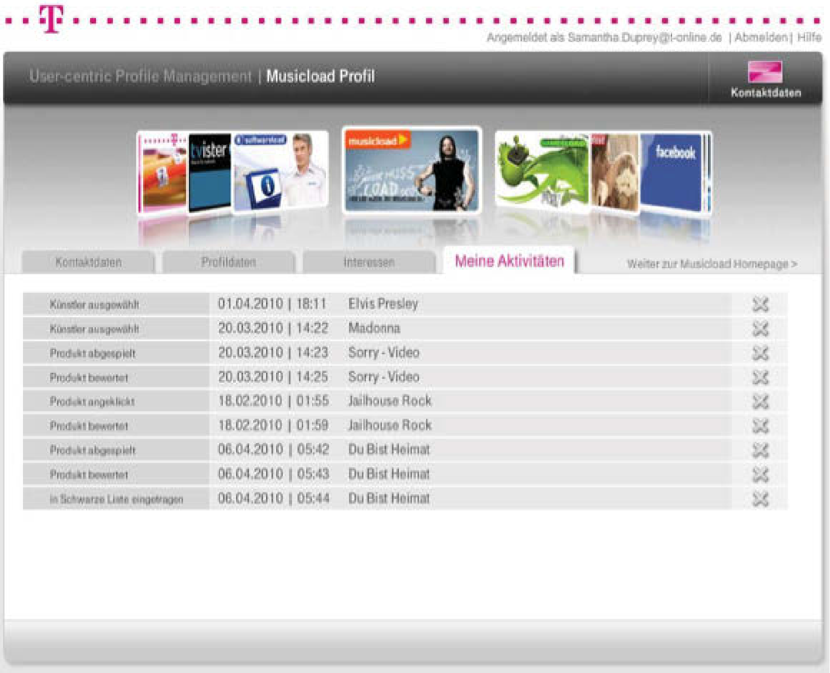
Ihr Profil ist bis 15. 05. 2010 im UCPM-Zwischenspeicher abgelegt.

Figure 4.7: Main UI showing different applications and personal information.

Fig. 4.8 gives an example of the type of information and how it is visualized in our system. The user has different information cards visualizing information stored about her. The user can see her last actions in an application, as an example of implicit information visualized by our system.

Privacy aware data sharing

The requirements for a privacy-aware approach for sharing information safely across applications is to make sure that no personal user information is shared


Angemeldet als Samantha.Duprey@t-online.de | Abmelden | Hilfe
User-centric Profile Management | Musicload Profil
Kontaktdaten
Kontaktdaten
Profildaten
Interessen
Meine Aktivitäten
Weiter zur Musicload Homepage >

Künstler ausgewählt	01.04.2010 | 18:11	Elvis Presley
Künstler ausgewählt	20.03.2010 | 14:22	Madonna
Produkt abgespielt	20.03.2010 | 14:23	Sorry - Video
Produkt bewertet	20.03.2010 | 14:25	Sorry - Video
Produkt angeklickt	18.02.2010 | 01:55	Jailhouse Rock
Produkt bewertet	18.02.2010 | 01:59	Jailhouse Rock
Produkt abgespielt	06.04.2010 | 05:42	Du Bist Heimat
Produkt bewertet	06.04.2010 | 05:43	Du Bist Heimat
in Schwarze Liste eingetragen	06.04.2010 | 05:44	Du Bist Heimat

Figure 4.8: Visualization of past behavior.

unintended. To fulfill these requirements, we build upon OpenID[5]. OpenID offers an interface to give permissions to third-party applications to use data and to actually share it. We have chosen OpenID as it is a well established technology supported by companies like Google[6] and Microsoft[7] and it had proven its applicability in other use cases [54]. The UI designed to support users in a privacy-aware use-cases is shown in Fig. 4.9. It allows users to control what kind of data is distributed to whom.

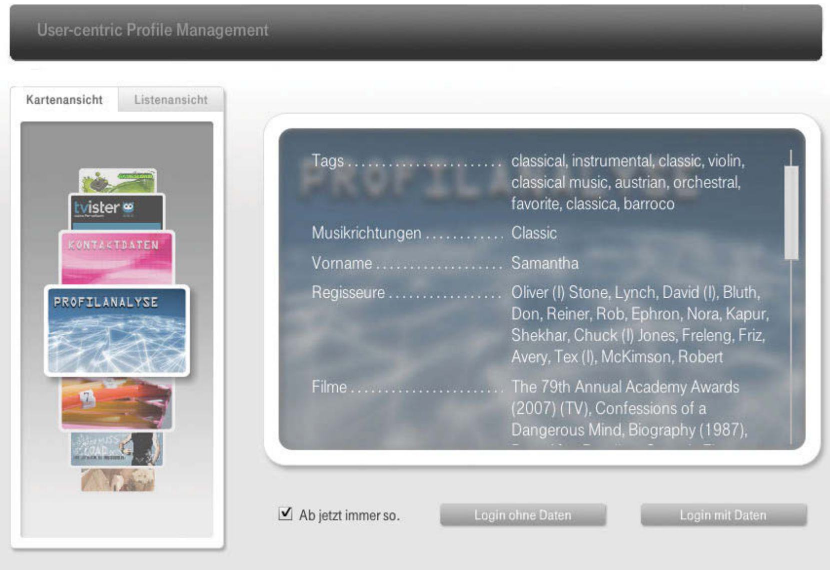
The actual data access is handled by an API which offers methods to request information on behalf of the user. Such a request can come from the user, who wants to access personal data or other systems that want to use the data for personalization or adaptation purposes. All data access must be confirmed explicitly by the user using the OpenId interface. If a system asks for information about 'AM#music_favorite_genres', and the user approves the request, the system gets the information stored in the musicApp attribute 'musicApp#fav_music_genres', the Facebook attribute 'facebook#music' and

[5] http://openid.net/

[6] http://www.google.com

[7] http://www.microsoft.com

the OtherApp data from 'otherApp#music_genres' as these are related to the 'AM#music_favorite_genres' field in the aggregation model.

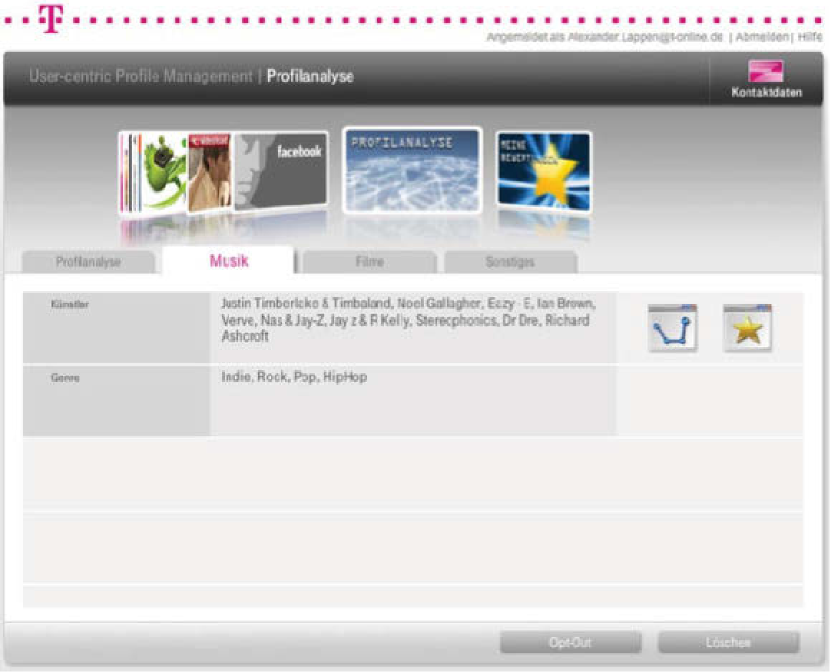


Tags	classical, instrumental, classic, violin, classical music, austrian, orchestral, favorite, classica, barroco
Musikrichtungen	Classic
Vorname	Samantha
Regisseure	Oliver (I) Stone, Lynch, David (I), Bluth, Don, Reiner, Rob, Ephron, Nora, Kapur, Shekhar, Chuck (I) Jones, Freleng, Friz, Avery, Tex (I), McKimson, Robert
Filme	The 79th Annual Academy Awards (2007) (TV), Confessions of a Dangerous Mind, Biography (1987),

Figure 4.9: Our OpenID interface.

Profile Enrichment

With "My Personal User Interface" we not only want to show that information can be easily aggregated with our ontology but also generate an additional value - for the user and other applications. Szomszor [199] showed that aggregation of different applications, from the tagging domain, lead to richer interest profiles. The 'Profile Analysis' card presents such new interests generated from all known interests information derived from the connected applications and also utilizing information from the Semantic Web. The algorithmic background is described in detail in Chapter 5. The idea is to create an enriched interest profile, with more preferences available, than only the information extracted from the connected applications. In Fig. 4.10 information that is implicitly inferred from existing user data, is shown. It shows new music artists (Künstler) and genres that are computed based on information extracted from the connected Facebook, Musicload and other applications. This information is intended to bootstrap the knowledge about a user if only few data is available. Within "My Personal User Interface" we build a Profile Analyzer component using information from the aggregated profiles and the Semantic Web. The complete enrichment process and an evaluation how this improves recommendation quality is given in Chapter 5.


Angemeldet als Alexander.Lappen@t-online.de | Abmelden | Hilfe
User-centric Profile Management | Profilanalyse
Kontaktdaten
facebook
PROFILANALYSE
MEINE EMPFEHLUNGEN
Profilanalyse
Musik
Filme
Sonstiges
Opt-Out
Löschen

Künstler	Justin Timberlake & Timbaland, Noel Gallagher, Eazy - E, Ian Brown, Verve, Nas & Jay-Z, Jay z & R Kelly, Stereophonics, Dr Dre, Richard Ashcroft
Genre	Indie, Rock, Pop, HipHop

Figure 4.10: Visualization of recommendations.

If a third-party application requests data of the user, the user is asked to give permission for that application to use the data. Therefore, the UI presents data that will be sent to the user. The process of sharing information is two-folded. After a third-party application requests data, e.g. about the user's musical taste, the system selects all information previously aggregated using the ontology. So, for a music taste request, data from Facebook or the previously described 'Profile Analysis'-profile would be selected. The second step is the validation through the user. Therefore, the selected information is presented to the user using the same card metaphor but only showing the information to be sent to the third-party application. The user can navigate through the different cards, see which information will be sent and decide to accept or deny the request. This makes it possible to easily see and control what data is shared.

4.2.3 Summary of user model aggregation

In this section, we introduced a use case to aggregate, access and manage personal information in a secure way. We presented a new ontology that defines a meta-model and supports the aggregation of distributed user models and a system that utilizes the ontology and allows users to fully profit from the semantic technology and to keep an overlook over their personal data secure data sharing by using OpenID. With My Personal User Interface and the 'Profile Analysis'-profile we also showed that the aggregated information could be used to create additional information. This information can be used to improve personalization and adaption and help users to profit more from such an personalized system. Of course, the user stays in control and can decide to discard this automatically added information. The system has been implemented in collaboration with the Telekom Innovation Laboratories[8]. The large scale use of the system depends on strategic decisions of their management. While this section focused on the first strategy of standardization approaches, the next section focuses on the second strategy, a common standardized user model.

4.3 The Semantic Web User Model

In the previous section, we presented a new meta ontology for aggregating user profiles from different applications following the aggregation approach described in Section 4.1. In the next sections, we present a standardized model for user model for the Social Web following the the standardization approach described in Section 4.1.

Every day, people in the Social Web create 1.5 billion pieces of information on Facebook, over 140 million tweets on Twitter, upload more than 2 million videos on YouTube and around 5 million of images to Flickr[9]. This huge amount of social data attracts researchers who want to use it to learn more about user preferences and interests, and enhance recommendation and personalization systems. Abel et al. showed in [3] that collecting information

[8] `http://www.laboratories.telekom.com`, Telekom Innovation Laboratories are the central research and development unit of the Telekom, located at Technische Universität Berlin.

[9] http://www.scribbal.com/2011/04/infographic-how-much-daily-content-is-published-to-twitter-facebook-flickr/

from different applications can improve completeness of data about a user. What most current systems have in common is that they use data from a single application and depend on sufficient user information (user behavior or ratings) to produce good results [8, 36]. By using the distributed personal information a single user produces on a daily base, and by building a holistic model of the user, personalization and recommendation quality can be further enhanced. But, for this holistic model the distributed user data has to be aggregated across applications. This idea is not new, it has existed since the 90's where different research initiatives proposed generic user modeling servers that build a central structure to manage and share user information [114, 120]. These approaches could not succeed because of their static, predefined user models while application-based user models strongly differ in the information they need to know about a user (as we will show in Section 4.3.1). Another reason for the failure was that applications do not want to lose control over their data, thus, a central storage was not wanted. New trends from the Semantic Web can provide a remedy. Instead of having a central server, ontology based user models are proposed to support data aggregation and sharing. Thus, applications can keep their data but use a common 'language' to model the information. While semantic technologies help to overcome technical problems, the main questions remain: What user information must a semantic model contain with focus on the Social Web? What requirements must a model fulfill to support data sharing and aggregation?

In this chapter, we want to give answers to these questions by analyzing user models from different Social Web applications and draw conclusions about the diversity and type of user information that such a generic user model should have. We discuss existing work and motivate a semantic Semantic Web User Model (SWUM). Requirements and structure of SWUM will be introduced in Section 4.3.1, it is based on the extensive analysis of 17 Social Web applications. The SWUM ontology itself is introduced and explained in detail in Section 4.3.2. In Section 4.3.3 we also carefully investigate what is needed to enable an easy, automated, aggregation process. To give a better understanding of the intended use of the SWUM we present a use case in Section 4.3.4.

The main contributions of this chapter are an extensive analysis of requirements of today's Social Web applications regarding stored user data and the introduction of a new Social Web user model that is:

- generally adapted to the needs of Social Web applications and
- that allows an easy data sharing between applications.

This work is intended to simplify the user model aggregation process by pointing out user information managed by Social Web applications and introducing a unified model as the basis for such aggregation processes.

4.3.1 Requirements for a Semantic User Model for the Social Web

To define a user model for the domain of the Social Web, we first have to understand the demands of social web applications on user models. Therefore, we did an extensive survey of the modeled user information of 17 well-known Social Web applications. The list of analyzed applications is shown in Table 4.2. The applications were chosen because of their size and level of awareness (number of users, global distribution). To be able to consider local differences, we also included applications that are strong in only one or two regions (Orkut in South America, Lokalisten and StudiVZ in Germany). We also selected Social Web applications from different kinds of domains, photo- and video-sharing platforms, short-message services, social networks, etc. To decide if the user information stored by an application is of importance, we picked at least two Social Web applications from the same domain.

Facebook	`http://www.facebook.com`	Myspace	`http://www.myspace.com`
Windows Live	`http://home.live.com`	YouTube	`http://www.youtube.com`
Flickr	`http://www.flickr.com`	Yahoo	`http://de.yahoo.com`
Picasa Web	`http://picasa.google.com`	StudiVZ	`http://www.studivz.net`
Digg	`http://www.digg.com`	Yelp	`http://www.yelp.com`
Lokalisten	`http://www.lokalisten.de`	Orkut	`http://orkut.com`
Identi.ca	`http://identi.ca`	LinkedIn	`http://www.linkedin.com`
Vimeo	`http://www.vimeo.com`	Xing	`http://www.xing.com`
LastFM	`http://www.last.fm`		

Table 4.2: List of 17 social applications that we analyzed for the requirements analysis

For each evaluated application, we collected the type of information and the internal attribute name. Table 4.3 shows the type of user information and where the information was found on the webpage. The internal attribute names, used by each application are particularly important as they are later used to define and name the attributes of the Semantic Web User Model (SWUM).

IU Name	Source code ID	Found on
Name	name	Registration Page
Firstname	firstname	Registration Page
Surname	secondname	Registration Page
Gender	gender	Registration Page
Birthday	birthdaygroup	Registration Page
Country	country	Registration Page
Postal Code	postalcode	Registration Page
Yahoo! ID and Email	yahooid	Registration Page

Table 4.3: Evaluation example for Yahoo: User information, attribute name and where the information was found on the webpage.

To be able to create our SWUM, we first have to decide which type of information, which user model dimensions, should be part of the model and which attributes in the different dimensions should be supported.

Model Dimensions

After collecting all the information, the first step is to determine the user model dimensions that our user model has to cover. As shown in GUMO, a lot of dimensions exist, but not all of them are required in the context of the Social Web. Several dimension are mentioned and discussed in the literature. We presented a consolidated taxonomy in Chapter 2.4 that bases on [192, 88, 103, 114, 50] and builds the basis for the selection of needed dimensions for our model. The following enumeration gives a short recapitulating overview of the dimensions.

- *Personal Characteristics* (or Demographics) range from basic information like gender or age to more social ones like relationship status.
- *Interests and Preferences* in an adaptive system usually describe the users' interest in certain items. Items can be e.g. products, news or documents.

- *Needs and Goals*: When using computer systems, users usually have a goal they want to achieve. Such goals can be to satisfy an information need or to buy a product. The plan to reach such goals is for example to support users by changing navigation paths or reducing the amount of information to a more relevant subset.

- *Mental and Physical State* describe individual characteristics of a user like physical limitations (ability to see, ability to walk, heartbeat, blood pressure, etc.) or mental states (under pressure, cognitive load).

- *Knowledge and Background* describe the user's knowledge about a topic or system. It is used in educational systems to adapt the learning material to the knowledge of a student, display personalized help texts or tailor descriptions to the technical background of a user. The knowledge and background is a long-term attribute on the one hand but can differ and change from session to session depending on the topic. Knowledge and background about certain topics can increase or decrease over time [50].

- *User Behavior*: The observation and analysis of user behavior is usually a preliminary stage to infer information for one of the previous mentioned dimensions. It can also serve for direct adaptation like using interaction history to adapt the user interface to common usage patterns of the user.

- *Context*: In computer science context generally refers to 'any information that can be used to characterize the situation of an entity' [67], but the discussion about what context actually is, is still ongoing [69]. In the area of user modeling, the term context focuses on the user's environment (e.g. location or time, or devices the user interacts with) and human characteristics. Human characteristics describe social context, personal context and overlap with the *Mental and Physical State* dimension).

- *Individual Traits* refer to a broad range of user features that define the user as an individual. Such features can be user characteristics like introvert or extrovert or cognitive style and learning style.

Based on this user taxonomy, we checked for all 17 applications if they cover these dimensions. Fig. 4.11 shows that social applications only cover some dimensions. All of the applications maintain *Personal Characteristics* and most of them also use *Interests and Preferences* information. Not used at all are the dimensions *Individual Traits* and *Mental and Physical State* which are more used in educational systems than in Social Web applications [50].

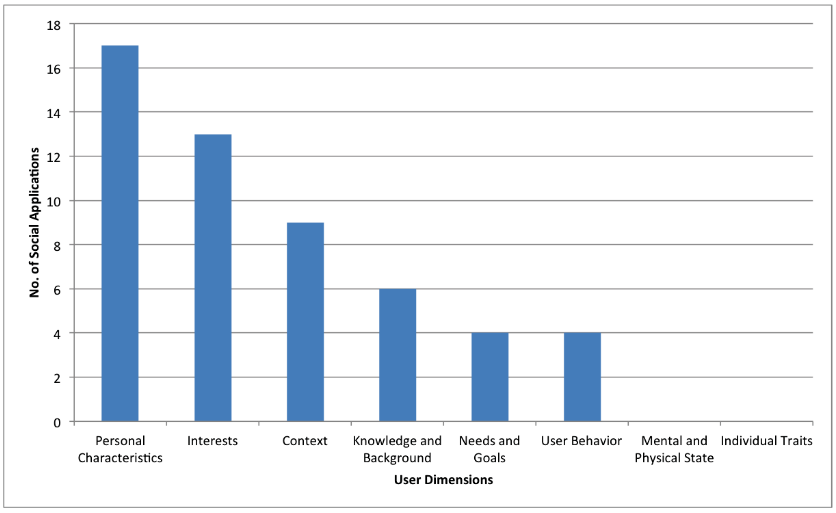


Figure 4.11: Number of applications storing user information in the different user dimension categories.

The usage of *Knowledge and Background* and *Context* depends on the focus of the social application. Social business applications, like LinkedIn or Xing, support the *Knowledge and Background* dimension as users can enter their college degree, areas of profession, etc. The support for the dimension *User Behavior* is not easy to work out, as user behavior usually is an implicit feature and not displayed on the user profile page of an application. It can be assumed, though, that almost all applications track user behavior on their site. A positive exception is 'Google Dashboard'[10] where a user gets an easy overview of the stored personal information e.g. previous search behavior. The *User Behavior* dimension, although it is an important piece of adaptation and personalization, is to complex to be part of a generic Social Web user model. For this purpose we recommend a specialized approach with an extra user behavior ontology as discussed in [165, 159] and presented in Chapter 3. *Context* is an important area as the latest research shows and of importance for a Social Web user model [7]. However, not all forms of context can be considered as a part of a Social Web user model. The analysis showed that the social context and location is of importance and therefore these sub-dimensions of context are part of SWUM. The importance of the context Time also seems of interest, but did not show up in our analysis.

[10] https://www.google.com/dashboard

From this analysis it follows that a main requirement for Social Web user model is, that it has to cover the user dimensions *Personal Characteristics, Interests, Knowledge and Behavior, Needs and Goals and Context (Social Context, Location)*. Accordingly, these dimensions are part of our SWUM.

User Model Attributes

After selecting the dimensions to be covered, we define the attributes that the user model should support.

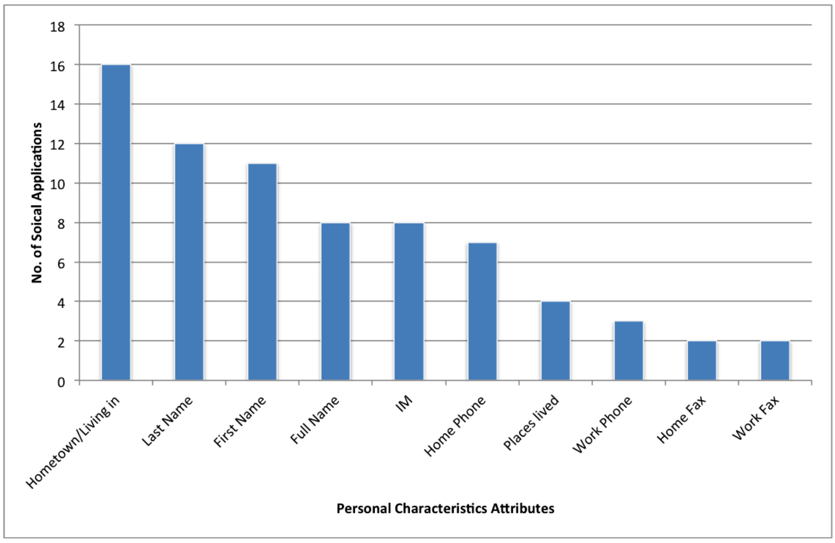
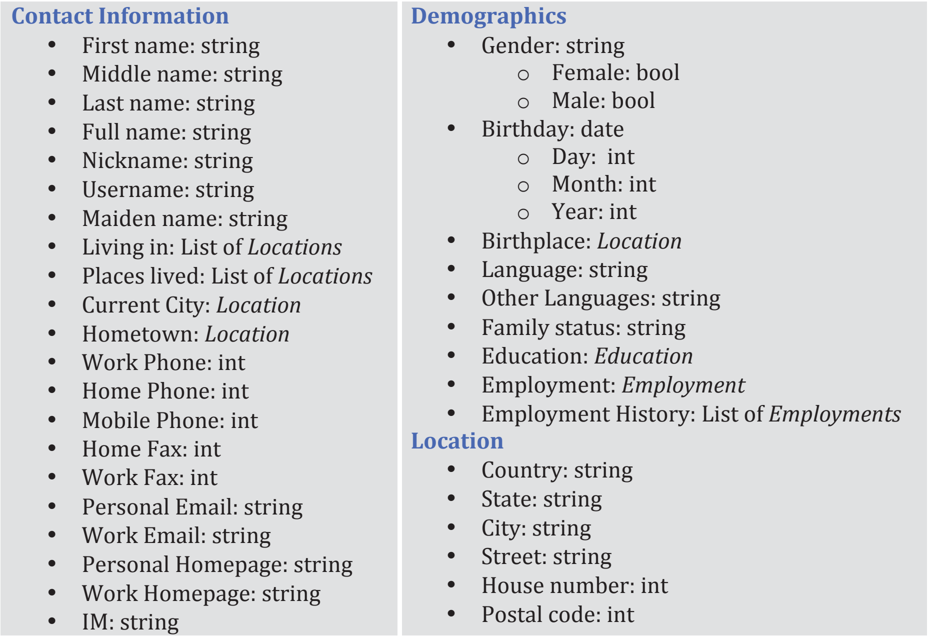


Figure 4.12: Attributes of the *Personal Characteristic* dimension and how often they occur in the different applications.

The procedure for the attribute selection is similar to the procedure used to select the dimensions. We checked the different attributes of the different applications. Fig. 4.12 gives an example for the *Personal Characteristic* dimension. It shows an excerpt of the attributes and how often they occur in the analyzed social applications. In this way, we selected a set of attributes for each dimension. An example for the *Personal Characteristic* dimension is shown in Fig. 4.13. The *Personal Characteristic* is divided into two main concepts namely Demographics and Contact Information. The concept Location is a helper concept to model locations and link certain information, e.g. places lived, to it.



Contact Information

- First name: string
- Middle name: string
- Last name: string
- Full name: string
- Nickname: string
- Username: string
- Maiden name: string
- Living in: List of *Locations*
- Places lived: List of *Locations*
- Current City: *Location*
- Hometown: *Location*
- Work Phone: int
- Home Phone: int
- Mobile Phone: int
- Home Fax: int
- Work Fax: int
- Personal Email: string
- Work Email: string
- Personal Homepage: string
- Work Homepage: string
- IM: string

Demographics

- Gender: string
  - ○ Female: bool
  - ○ Male: bool
- Birthday: date
  - ○ Day: int
  - ○ Month: int
  - ○ Year: int
- Birthplace: *Location*
- Language: string
- Other Languages: string
- Family status: string
- Education: *Education*
- Employment: *Employment*
- Employment History: List of *Employments*

Location

- Country: string
- State: string
- City: string
- Street: string
- House number: int
- Postal code: int

Figure 4.13: SWUM attributes for *Personal Characteristic* dimension.

In the following section, we will give a detailed overview and description of the resulting user model.

4.3.2 Model description of the Semantic Web User Model Ontology

The SWUM is a collection of different linked entities that give a complete picture of the features needed to cover all aspects of the Semantic Web. A complete overview of the SWUM is given in Fig. 4.14.

In the following, we introduce the different entities that form the SWUM, describe the data properties and the intended usage. The entities and data properties are results of the previously described analysis of current Social Networks and thus build standardized user model suitable for the semantic web. The SWUM ontology is accessible through [154]. The ontology description is OWL DL compliant.

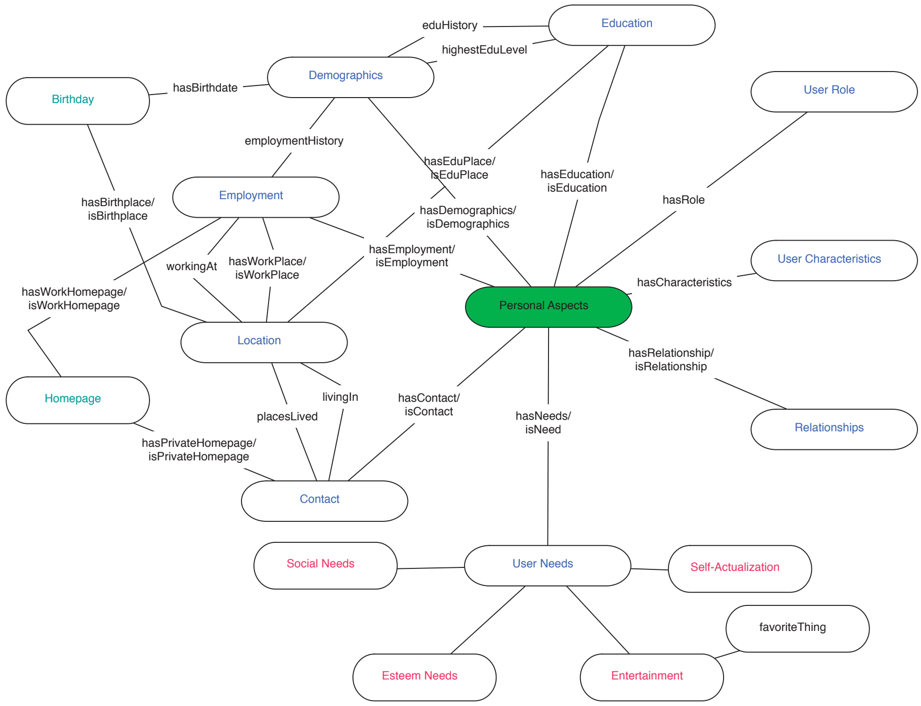


Figure 4.14: The Semantic Web User Model Ontology with all entities and links between them.

Personal Aspects

The Personal Aspects entity connects all different entities representing the user model dimensions identified (in Section 4.3.1) to be important for a Semantic Web User Model. The *Personal Characteristics* dimension is covered by the entities contact, demographic and location information. The *Interests* and *Needs* dimension is reflected in the entity User Needs and its sub-entities Social Needs, Esteem Needs, Self-Actualization and Entertainment. The entities employment and education history describe the *Knowledge* dimension. Context, as said, is currently not fully covered, only Social Context is included in the Social and Esteem Needs entities. In Table 4.4 we explain all relations properties of the SWUM. Following that, we describe all entities and its data properties.

SWUM Object Properties		
Predicate	Inverse Direction	Description

swum:eduHistory	–	This connects a person with all Education entities of that person.
swum:highestEduLevel	–	This connects a person with the Education entity that marks the current highest degree of that person.
swum:hasEduPlace	swum:isEduPlace	This property connects an Education entity with a Location entity. The inverse links an Place to an Education entity.
swum:hasDemographics	swum:isDemographics	This property connects the Demographic information with a person.
swum:hasBirthday	–	This property creates a link between a Birthday and a Demographic entity.
swum:hasBirthplace	swum:isBirthplace	This property defines a Location as the Birthplace of a person.
swum:hasEducation	swum:isEducation	This creates direct links between a Personal Aspect entity and a corresponding Education entity.
swum:hasRole	–	This property connects the Role information with a person.
swum:hasCharacteristics	–	This property connects the Characteristics information with a person. s
swum:hasNeed	swum:isNeed	This property connects the Needs information with a person.
swum:hasContact	swum:isContact	This property connects the Contact information with a person.
swum:hasEmployment	swum:isEmployment	This property connects the Employment information with a person.

swum:placesLived	swum:livingIn	This connects a person with all Locations a person previously lived at.
swum:hasWorkPlace	swum:isWorkPlace	This property connects a Employment and a Location entity.
swum:workingAt	–	This property defines the current working place of a person and connect a Person with an Employment entity.
swum:employmentHistory	–	This connects a person with all previous working places (Employment entities) of that person.
swum:hasPrivateHompage	swum:isPrivateHomepage	This property connects a Homepage entity with the Contact entity.
swum:hasWorkHomepage	swum:isWorkHomepage	This property connects a Homepage entity with the Employment entity.

Table 4.4: The Object Properties of the SWUM Ontology.

Contact

The *Contact* entity describes personal user information such as the name or age. It also links to locations such as the home or working address. Fig. 4.15 shows the different data properties of the Contact entity.

The properties are described in detail in Table 4.5.

Contact Data Properties		
Predicate	Object Type	Description
swum:givenName	rdfs:Literal	This property defines the first name of a user.

swum:middleName	xsd:Literal	This property allows to define a middle name of a user
swum:familyName	xsd:Literal	Last name (surname) of a person
swum:fullName	xsd:Literal	First and surname of a person.
swum:nickname	xsd:Literal	This property defines
swum:fax	xsd:Literal	Fax number of a person or of a work place.
swum:telephoneNumber	xsd:Literal	Telephone number of a person. Private or work related.
swum:maidenName	xsd:Literal	Birth name of a person.
swum:mail	xsd:Literal	This property defines the email address.
ubo:mobileNumber	xsd:Literal	The mobile phone number of a person.

Table 4.5: Properties of the OWL class **Contact**.

The Homepage entity is connected to the Contact entity and the Employment entity.

Homepage Data Properties		
Predicate	Object Type	Description
swum:url	xsd:Literal	Defines the URL of the page.
swum:title	xsd:Literal	Short name for a homepage.

Table 4.6: Properties of the OWL class **Homepage**

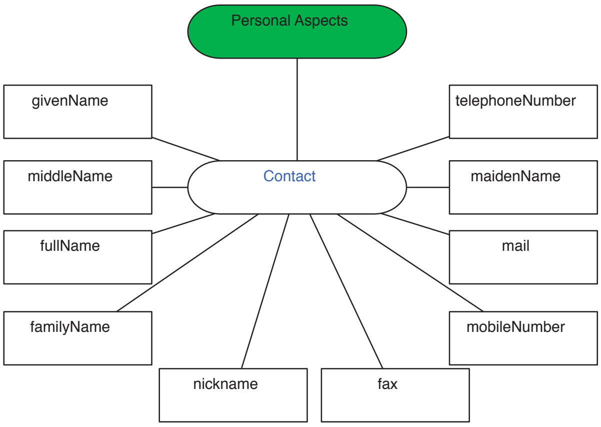


Figure 4.15: The Contact entity with data properties.

SWUM Demographics

This entity covers statistical information about the user such as gender and languages the user speaks. The SWUM Demographic entity covers only few aspects as often also characteristics such as race or disabilities are included. In social networks however, such information is not of relevance.

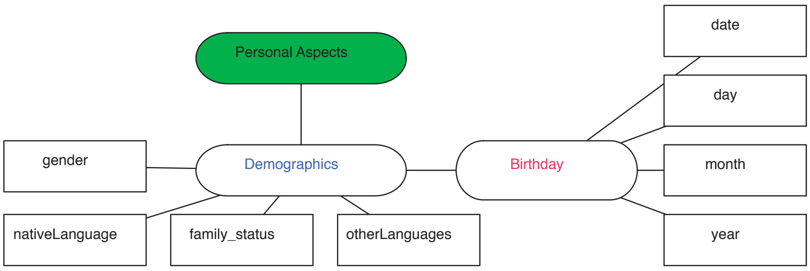


Figure 4.16: The Demographics entity with attributes.

Demographics Data Properties		
Predicate	Object Type	Description
swum:gender	rdfs:Literal	This property defines the gender of a person, usually ”male” or ”female” plus ”other”.

swum:family_status	xsd:Literal	Marital status of a person. Possible answers are ”single” or ”married”, ”divorced”, ”widowed” etc.
swum:nativeLanguage	xsd:Literal	Mother tongue of a person.
swum:otherLanguages	xsd:Literal	This property defines languages the user can speak but that are not his/her mother tongue.

Table 4.7: Properties of the OWL class **Demographic**

Birthday Data Properties		
Predicate	Object Type	Description
swum:date	xsd:Literal	This property defines a birthday of a person.
swum:day	xsd:Literal	Defines the day of a birthday.
swum:month	xsd:Literal	Names the month of the birthday.
swum:year	xsd:Literal	The year of the birthday.

Table 4.8: Properties of the OWL class **Birthday**

SWUM Education

A person’s educational history can be modeled with the Education entity, see Fig. 4.17.

Education Data Properties

Predicate	Object Type	Description
swum:educationLevel	rdfs:Literal	This property defines the reached graduation. Master, Bachelor etc.
swum:educationName	xsd:Literal	This property defines the name of the educational organization.
swum:educationDescription	xsd:Literal	This property allows to describe the type of school: high school, university etc.
swum:educationPeriod_start	xsd:Literal	This property defines the start date of the education.
swum:educationPeriod_end	xsd:Literal	End date of the education.

Table 4.9: Properties of the OWL class **Education**

SWUM Employment

The Employment entity describes work related information and is especially in business social networks important. Fig. 4.18 shows the entity, Table 4.10 explains the data properties. A person can have more than one Employment entity that represents a current position.

Employment Data Properties		
Predicate	Object Type	Description
swum:companyName	rdfs:Literal	This property defines the first name of a user.
swum:telephoneNumber	xsd:Literal	This property allows to define a middle name of a user
swum:mobileNumber	xsd:Literal	This property defines a number of a work related mobile phone

swum:fax	xsd:Literal	This property defines the number of a fax machine.
swum:workPosition	xsd:Literal	This property allows to set the title of the current job position. E.g. project manager, developer, etc.
swum:workPeriod_start	xsd:Literal	This property defines the beginning of a period where a person was employed at a company.
swum:workPeriod_end	xsd:Literal	This property defines the end of employment period. An employment must have a beginning but don't need an end.
swum:mail	xsd:Literal	This property defines a work related Email address.
swum:salary	xsd:Literal	This property defines the salary the person earned for the job.

Table 4.10: Properties of the OWL class **Employment**

SWUM Location

The Location entity, Fig. 4.19, describes a physical location. A Location in the SWUM Ontology can be the birthplace of a person, the address of e.g. the school or university, or the location of the work place. The data properties of the Location are described in Table 4.11.

Location Data Properties		
Predicate	Object Type	Description
swum:postalCode	rdfs:Literal	This property defines the postal code of a location.
swum:houseNumber	xsd:Literal	Street number of the house/place.

swum:street	xsd:Literal	Name of the street of a Location.
swum:city	xsd:Literal	The city, usually the name of the city, of a Location.
swum:state	xsd:Literal	Name of the state of a Location.
swum:country	xsd:Literal	This property defines the name of the country of a Location.

Table 4.11: Properties of the OWL class **Location**

SWUM Needs

The User Needs entity, shown in Fig. 4.20, covers the dimensions *Interests* and *Needs and Goals*. It consists of four different sub-entities - Social Needs, Esteem Needs, Self-Actualization Needs and Entertainment. The first three mentioned entities are named based on Maslow's hierarchy of needs [133].

- **Social Needs**: Describes interpersonal needs, feelings. The data properties are described in Table 4.12.

Social Needs Data Properties		
Predicate	Object Type	Description
swum:intimacy	rdfs:Literal	This property describes whether a person has the need for sexual intimacy with other persons.
swum:family	xsd:Literal	Describes whether a person is a family man or not.
swum:friendship	xsd:Literal	This property defines whether a person is interested in social relationships with other people.

swum:communication	xsd:Literal	This property describes the need of a person for communication with other persons. It contains information about whether the person needs communication or not.

Table 4.12: Properties of the OWL class **Social Needs**

- Esteem Needs: Esteem presents the normal human desire to be accepted and valued by others, see Table 4.13.

Esteem Needs Data Properties		
Predicate	Object Type	Description
swum:respectByOthers	rdfs:Literal	Describes whether a person needs to be respected by others or not.
swum:respectOfOthers	xsd:Literal	This property defines whether a person respects others or not.
swum:self-esteem	xsd:Literal	This property describes whether a person has self-esteem or not.
swum:achievement	xsd:Literal	This property allows to define whether a person has the need for achievement or if this need is already satisfied
swum:self-respect	xsd:Literal	Describes whether a person has self-respect or not.
swum:confidence	xsd:Literal	The property describes whether a person has a lack of confidence or not.

Table 4.13: Properties of the OWL class **Esteem Needs**

- Self-Actualization Needs: Maslow describes this need as the need to

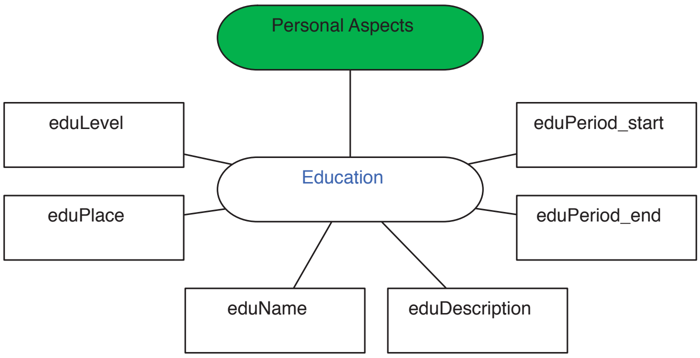


Figure 4.17: The Education entity with attributes.

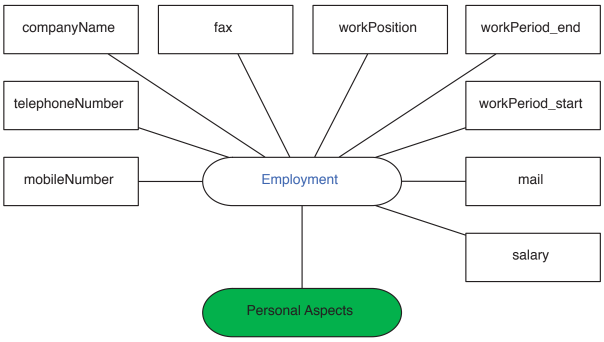


Figure 4.18: The Employment entity with attributes.

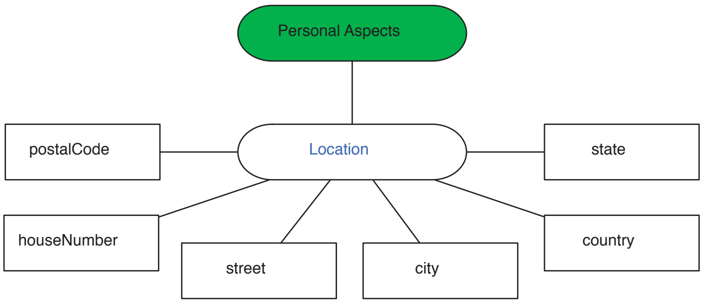


Figure 4.19: The Location entity with attributes.

become more and more what one is, to become everything that one is capable of becoming.

Self-Actualization Data Properties		
Predicate	Object Type	Description
swum:lackOfPrejudice	rdfs:Literal	This property allows to describe whether a person is narrow minded or not.
swum:morality	xsd:Literal	This property describes whether a person's character has the attribute to be ethical.
swum:creativity	xsd:Literal	This property defines whether a person's character has the attribute to be creative.
swum:spontaneity	xsd:Literal	This property describes whether a person's character has the attribute to be spontaneous.
swum:problemSolving	xsd:Literal	This property allows to describe how good a person is able to find solutions for a problem.
swum:AcceptanceOfFacts	xsd:Literal	This property describes whether a person is willing to accept facts or not.

Table 4.14: Properties of the OWL class **Self-Actualization**

- Entertainment: This describes the need for distraction, interests, etc. and covers the Interest dimension.

Entertainment Data Properties		
Predicate	Object Type	Description
swum:favoritePeople	rdfs:Literal	This property defines the people a person is a fan of, e.g. favorite actor, artist etc.
swum:activitiesInterest	xsd:Literal	This property allows to define favorite leisure activities such as playing football.

swum:favoriteThing	xsd:Literal	This property defines all things that a user likes, and that are not persons. For example the favorite music album or football club.

Table 4.15: Properties of the OWL class **Entertainment**

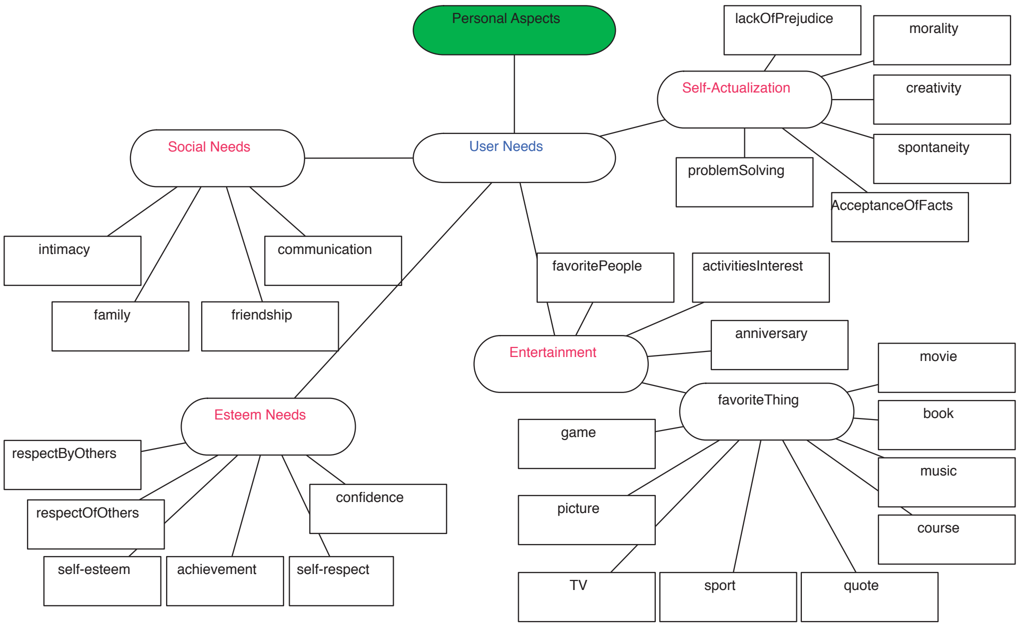


Figure 4.20: The Needs entity with attributes.

4.3.3 A User Model Word Net

An important outcome of the attribute distribution analysis was that often similar information is stored by most applications, but in differently named attributes, e.g. name (Yahoo) and real_name (LastFM) or homepage (LastFM) and website (Flickr). This problem of attribute name heterogeneity complicates a possible aggregation using a Meta-Model strategy. To cover that problem, we decided to extend our model with a WordNet like lexicon called User Model Word Net (UMWN). WordNet defines word sense relations between words. If a word represents a user attribute, the relatedness between different attributes can be acquired easily. However, many user attributes are not defined in WordNet. Moreover, many terms in WordNet are useless for user profile aggregation. Hence, the standard WordNet does not help, thus, we designed a reduced WordNet, specialized to serve the user profile aggregation and initially based on the attribute distribution of our analysis. The decision to use a WordNet based structure comes from the fact, that WordNet has a flexible and well-defined lexicon schema, which is publicly known and accepted. The user model terms can be linked to each other accurately by using the properties defined in WordNet. An example is depicted in Fig. 4.21 where the word sense relations for name and date are shown.

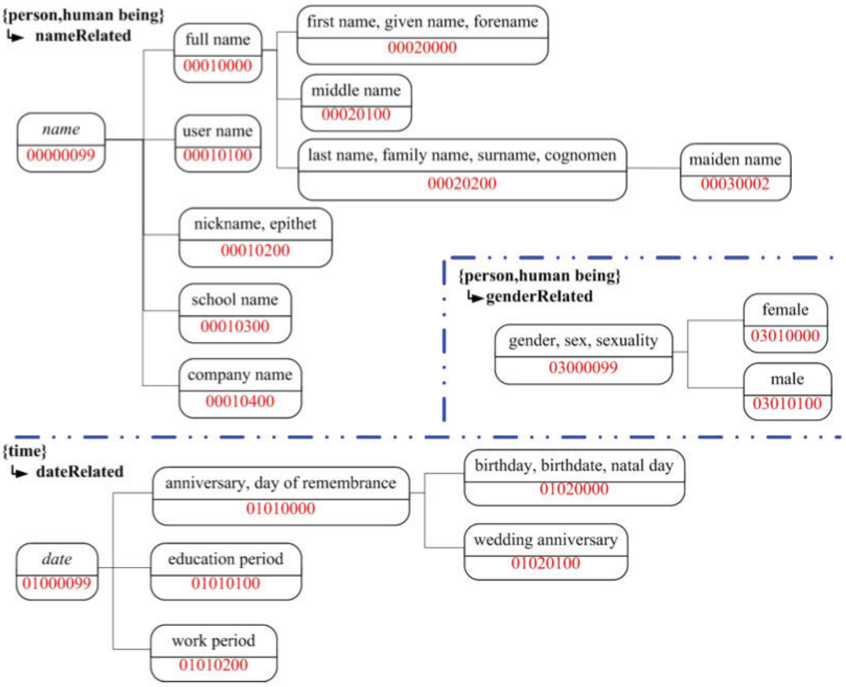


Figure 4.21: User Model WordNet relations.

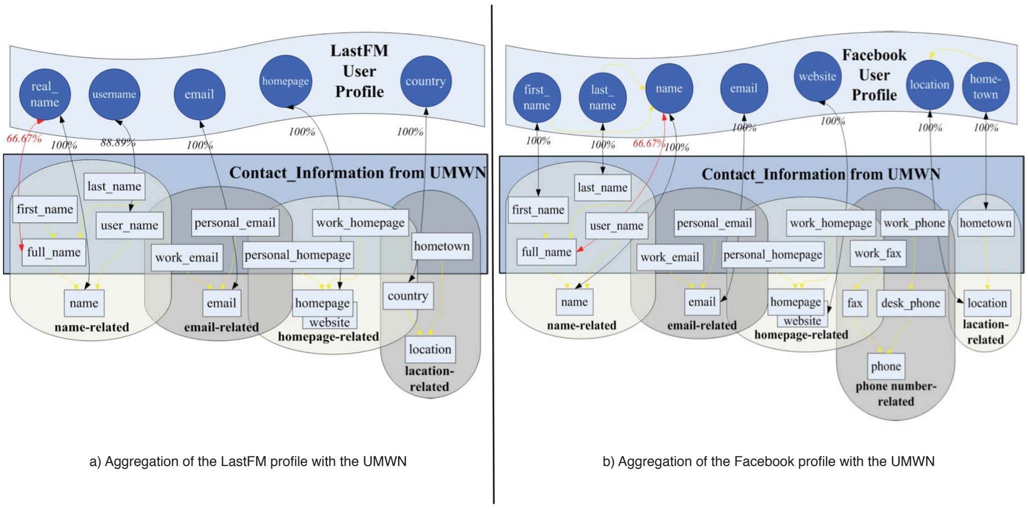
The UMWN is an important step for an automatized aggregation of different user models. It defines different types of word relations. The 'Name' concept describes the relations between different types of name attributes that can occur in a user model. The concept 'full name' consists of different subclasses like 'first name', which has several synonyms ('given name' or 'forename'). UMWN is stored in RDF(s)/OWL. Using ontology structures has the advantage that such a model is not static and can be easily extended. Our UMWN is extensible, towards not only the individuals, but also towards the schema of UMWN. Due to the highly distributed and heterogeneous user information in different user models, extensibility is an important feature. The UMWN contains currently ca. 520 syn sets where around 200 are unique in the User Model WordNet and not part of the common WordNet. It also contains over 100 antonyms and homonyms and 200 meronyms.

4.3.4 Use Case: Profile Aggregation with the SWUM

To outline the intended usage and functionality of the SWUM (which includes the UMWN) we want to exemplary explain the steps needed to aggregate a Facebook user model and a LastFM user model. The aggregation is a two-step process which we want to explain by the example of the website/homepage attribute shown in Fig. 4.22. The first step is to connect the LastFM attributes to the SWUM (see Fig. 4.22a). The LastFM user model has the attribute 'homepage' which can be directly linked to the SWUM, with a concept match of 100%. The Facebook profile (Fig. 4.22b) contains the attribute 'website' which is also part of our SWUM and thus, the attribute can also be linked to the SWUM without any extra effort.

The second step is then to directly connect the LastFM and Facebook user model as shown in Fig. 4.23. Based on the previously shown aggregation, connecting both models is straightforward. Revisiting the homepage/website example, these attributes can be directly linked because of the UMWN. The UMWN defines a synonym relation between the concepts 'homepage' and 'website', thus the LastFM and Facebook attribute can be directly linked with a match of 100%.

The aggregation of attributes that are not part of the SWUM can be done not only using the attribute name but also using the attribute content. So could an analysis show that the LastFM attribute 'real_name' often contains the users' full name and thus a connection with the SWUM/UMWN attribute 'full_name' can be done. Or the missing attributes can be added to the SWUM which is easy to do as it is a flexible RDF/OWL structure.



a) Aggregation of the LastFM profile with the UMWN

b) Aggregation of the Facebook profile with the UMWN

Figure 4.22: First step of the aggregation process. Figure a) shows how the attributes of LastFM and the SWUM/UMWN are connected. Figure b) depicts the connections of the Facebook profile.

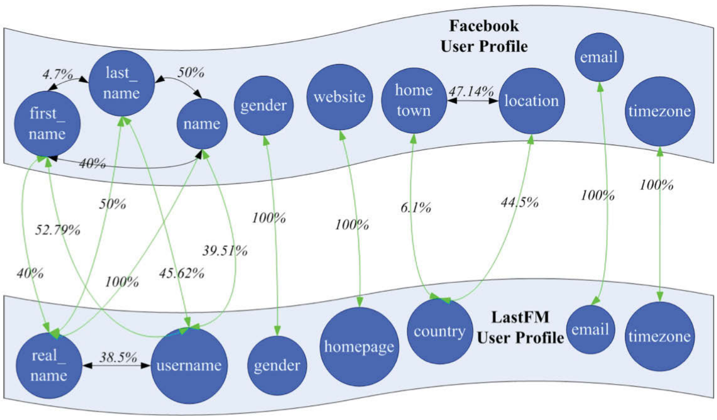


Figure 4.23: Aggregated LastFm and Facbook profiles.

4.3.5 Conclusion

In this section we introduced a new standardized user model, which fits the second strategy mentioned in Section 4.1. We explicitly wanted to answer the question what are the requirements of the Social Web for a user model to profit from the available distributed user information. We present the new user model, the Semantic Web User Model (SWUM) that is fitted to the needs of the Social Web. We therefore conducted an extensive analysis of 17 social applications and to specify requirements, which dimensions and attributes are needed, for a Social Web user model. Based on this analysis we defined the dimensions a Social Web user model must cover and explained how the decision process was conducted. The analysis showed that a Social Web user model only needs to cover certain dimensions of the user, namely *Personal Characteristics, Interests, Knowledge and Behavior, Needs and Goals and Context (Social Context, Location)*. We also presented the procedure to define the attributes of such a Social Web user model. To cover the problem of attribute heterogeneity throughout different social applications, we also equipped our model with a reduced WordNet that is especially tailored to the area of user modeling, the User Model Word Net (UMWN). The complete SWUM and UMWN model is based on RDF/OWL and thus easy to extend and reuse.

4.4 Closing Discussion

In this chapter we discussed the problem of distributed user information and approaches to build a common view on the data. We introduced a meta model, the Profile Data Model (PDM), to aggregate and weight information from different applications. The PDM serves as a way to connect data from existing applications, as shown in the UCPM project.

The Semantic Web User Model ontology (SWUM) is an approach to define a common model for applications to easy sharing and re-using user information. Based on the analysis of 17 different social web applications we defined the SWUM to fit the needs especially of the Social Web.

We show in the next chapter how semantics, user model ontologies and semantic knowledge in the world wide web, can improve adaptive systems. We present an evaluation on how recommendations can be improved utilizing semantic techniques.

Chapter 5

Evaluation of Semantic User Models for Recommendations

The flood of available information and products offered by Web applications like on-line retailers and news portals overwhelms today's users. To handle this information overload applications typically offer some kind of personalization techniques in most cases personalized filtering or personalized recommendations [193, 12]. However, personalized recommendations that adapt to the users' individual taste are a major challenge [8]. On the one hand, personalized recommendations improve user satisfaction and can motivate users to return. Bad recommendations on the other hand, may cause users to turn their back on those applications. A common recommendation approach is Collaborative Filtering (CF). CF utilizes historical user information, like ratings or interactions, to compute recommendations [198]. For users where the system has no or little information, like new users, user preferences need to be acquired first. This problem is known as the cold start problem [174]. The cold-start problem can be defined as the problem when a recommendation system does not have sufficient information about the past user preferences and rating behavior. One frequently used method to overcome the problem is to explicitly ask users to enter preferences or to show a selection of products the users should rate. This initial training phase of a recommender incurs additional effort for users and discourages them from using those applications, as users are not willing to spend a lot of time before they can profit from an application. Personalized recommendations help users to discover interesting information and products based on their preferences and tastes. In cases where no or only little information about the user is available, known as the grey sheep and cold start problem [51], recommendation quality is typically very low. In this chapter, we present a semantic approach to overcome the grey sheep and cold start problem by enriching the user profile with knowledge extracted from the Semantic Web. We explain the approach in detail and conduct a comprehensive evaluation to examine how the enrichment influences recommendation quality. Results show that our approach improves recommendation results especially for users with uncommon interests.

The chapter is structured as follows: We first give a short overview of the measures used to evaluate the performance of adaptive systems, precision, recall and F-Measure. The evaluation is done on a data set collected during the time of thesis writing. The data set bases on user information collected from FriendFeed[1], Facebook and LastFM[2] and will be explained in detail in Section 5.3.1.

The main contributions for this chapter have been published in [120, 128, 129].

5.1 Introduction to Evaluation and Enrichment

The Social and Semantic Web has attracted a large number of researchers from different research fields to find solutions to the cold start problem. So far, different approaches have been proposed. Approaches range from manipulating the CF process or manipulating the user model before the CF calculation. In the following section we will present selected works about State of the art CF systems that cope with the cold-start-problem and we present recent work about user profile enrichment.

Collaborative Filtering In [11] the authors present an approach that uses existing ontologies, e.g. a movie ontology, and integrate derived item information with existing user ratings. While standard CF algorithms assume that all items are distinct, the authors propose an extended CF algorithm that consider item information as well based on the item similarity, e.g.

[1] http://friendfeed.com

[2] http://last.fm

same director. Item similarity is computed by taking into account similarity between item attributes. To compute the attribute similarities, for each attribute a similarity function must be defined and an aggregation function that combines the different attribute similarities. In this way, it is possible to find similar users even if they did not rate the same, but similar movies. The approach has the disadvantage that it needs effort to build a similarity function for each attribute and it is also limited to one domain. With our approach we overcome both limitations of this work. Different weights for different relations/attributes can be learned automatically based on the number of occurrences in the graph for example, and the domain limitation is dropped because of our semantic approach where it is easily possible to bridge different domains.

In a different approach, Middleton et al. [139] build ontological profiles for users to recommend research articles. The user profile creation is done using a topic hierarchy. To overcome the cold-start problem, the authors also attempt to use externally available information based on personnel records and user publications. The limitation is that the existence of such additional knowledge cannot be generally assumed. In some cases, like the presented research community example, public information is available, but especially in the social web, this information is locked in the different social networks. Thus, instead of requiring personal information from external sources, our approach leverages public knowledge sources like Freebase (or DBpedia).

User Profile Enrichment Different strategies have been proposed to expand the knowledge about users ranging from the aggregation of user information distributed over different applications to solutions adding semantic and linguistic knowledge to user profiles [152, 121]. Aggregation of personal information from several applications [206, 163] and using it for recommendations has been demonstrated in experimental setups [27]. However, this approach is not easily adoptable as most applications keep their data in 'walled gardens' where the application provider does not allow to get any user information out of the system, e.g. no API is offered. Thus, it is not easy to get data for one user from different applications [27, 28]. In addition, privacy and security issues may occur and users may not be willing to share passwords to allow the aggregation of data from different accounts. Other works add meta-knowledge from sources like WordNet to user profiles to describe similar items, e.g. items from the same domain [123]. Of course, the aggregation of user information from different applications a user could help to build a holistic view of the user, but as the data is hard to get, we have chosen a more applicable way by using free encyclopedic data as the source for profile enrichment.

5.2 The Enrichment Approach

The general idea of our enrichment approach is visualized in Fig. 5.1 with an example of a music recommendation system: The figure shows three user profiles consisting of only a few items without any overlap with the other profiles. In this case, CF cannot be used as no similarities between items or users can be computed, which is needed for CF. Our profile enrichment process adds several new items (strongly related to the already present items), so that afterwards, the user profiles have an overlap and CF can be applied. If a user profile (middle row) initially contains user interests about 'Björk' and 'Moby', our enrichment algorithm takes both entities as an input and starts to traverse the semantic data set which is a graph where all information is connected. The first entity that is added to the user profile is the genre entity 'electronic', as both artists are directly connected to it. Then, the algorithm adds additional artists like 'Morcheeba' as the band is also connected to 'electronic'. This enriched user profile is then used for CF.

In this chapter we focus on music data to show and evaluate our approach. The approach itself presented in this section is designed to work on any kind of data as long as it is presented as a graph. Fig. 5.2 shows the general data structure needed for our approach. The data set needs a user node that is connected with a like/rated relation to a set of entities, which can be connected by any kind of relation. The rate/like relation indicates a positive relation to the linked entity. Negative relations are currently not considered. The entity nodes can be music information, as in our scenario, or books, movies etc.

Enriching user profiles based on semantic data

Our motivation is to cope with the cold-start problem. Therefore, we use semantic encyclopedic knowledge to extend small user profiles. Studies about

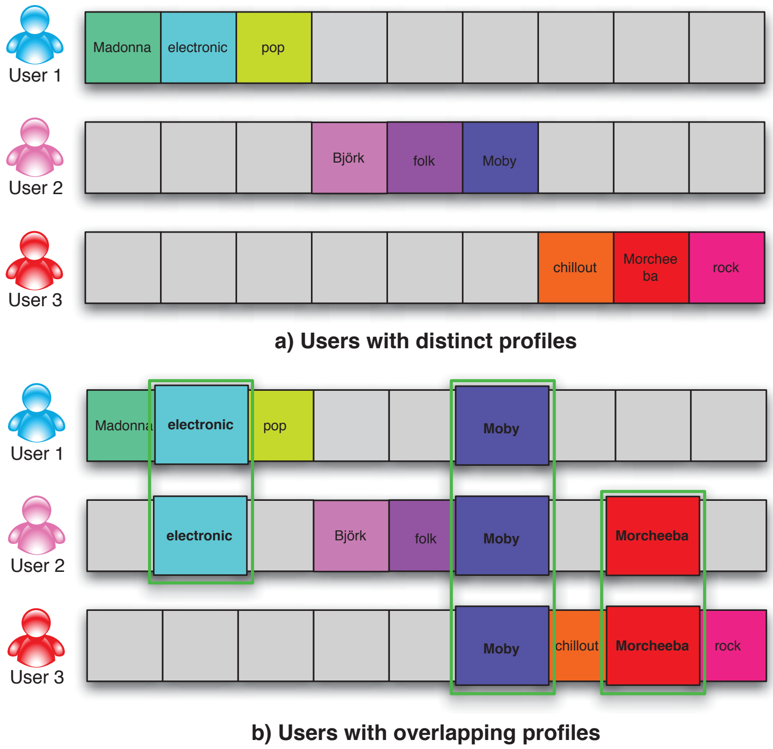


Figure 5.1: Simplified visualization of the initial cold start problem. a) Before the enrichment, there is no overlap between the different user profiles and collaborative filtering is not possible. b) After the enrichment, the user profiles overlap and collaborative filtering is possible

Wikipedia[3], as an example for online encyclopedias, proved that the quality and the accuracy of Wikipedia articles is on a high standard and hence a reliable information source [101]. Therefore, we follow the idea that semantic encyclopedic data is a good and 'neutral' source for enriching user profiles with knowledge not influenced by subjective opinions or tastes. Enriching user profiles with items strongly related to the items already present in the user profile, adds 'synonyms' for the existing entities. A synonym in this context means that we add interests to the user profile that are similar to already expressed user tastes, e.g., adding an additional artist that is related to an artist in the user profile. This is done to increase the overlap of the enriched user profile with other profiles. Thus, it improves the similarity calculation, but does not change the taste of the user.

[3] http://www.wikipedia.com

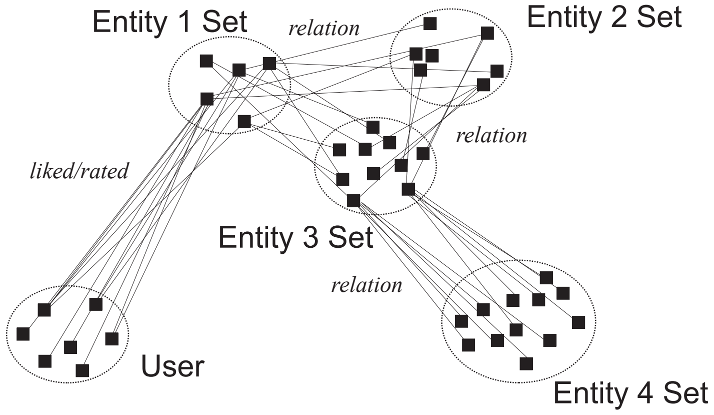


Figure 5.2: The semantic data set with generic information and user profiles linked to it.

Finding related items based on encyclopedic data

Our approach for solving the complex problem of computing entities to enrich the user profile uses link prediction methods on a semantic data set to find important related items to a given input set of items (e.g., a user profile). The link prediction task describes the problem of inferring missing links in an observed graph that are likely to exist [168, 200]. In our approach, we apply link prediction for the task of finding edges between items in the semantic data set and a set of given entities of a user profile.

To compute related entities for a given set of input items, we determine the entities best connected to the input entities already present in a user profile. In our scenario, *best connected* from a set of input entities describes the items that can be reached by several parallel paths each consisting of a small number of edges. The computation of the related entities can be performed directly on the semantic data set ("memory-based") or based on a simplified network model ("model-based"). The semantic data set is modeled as a network consisting of nodes representing the entities and edges describing the relationship between the entities (see Table 5.1). For computing entities closely related to a given user profile, we take all existing entries in the user profile as a starting point and traverse the semantic network ("path based breadth-first search"). Since an extensive search may require to much resources (CPU, RAM), we introduce a parameter to control the search depth of our approach. In this work, we used a maximum search depth of four, meaning that starting from the user profile all nodes are considered that can be reached with four steps or less. All entities that can be reached from entities in the user profile are weighted by the number of parallel paths and by the number of edges for each path. The formulas for calculating the path weights are shown in Fig. 5.3. An entity is the more relevant the more parallel paths from the user profile exist and the shorter (based on the number of edges) the paths are. Also the type of edge is taken into account. We evaluate for different path lengths how the profile enhancement influences the CF performance.

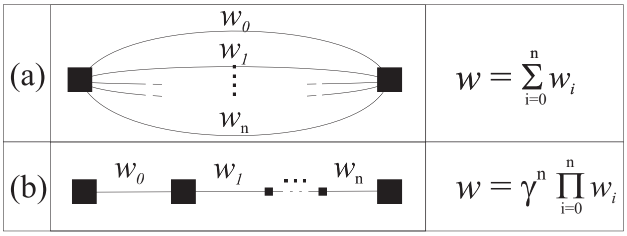


Figure 5.3: The figure shows the formulas for calculating the path weights for (a) parallel edges and (b) for a sequence of edges. The discount factor γ ensures that short paths get a higher weighting than long paths.

To give an impression how the system computes related entities, Fig. 5.4 and Fig. 5.5 show example computations using only artist and genre nodes and edges connecting those nodes. Fig. 5.4 shows a possible enrichment based on a user profile containing "Lady Gaga" as an interest. The path length is set to two, this means that only entities that are not more than two steps away are taken into account. In this example, "Madonna" would be used to enrich the user profile. Fig. 5.5 shows the enrichment going to a depth of four. This means that entities that are not more than four steps away are taken into account. Input is the same user profile, with "Lady Gaga" as an interest.

Memory-based link prediction We apply a path-based approach for computing predictions. Starting from several input entities (e.g. the entities in the user profile), we traverse the semantic network. The entities reachable from the input entities are ordered according to a semantic similarity rating. This rating is calculated based on the edge weights of the respective path. Currently, the weight of the edge, which can be considered as the importance of the edge, has to be set manually or by using normalization strategies. One

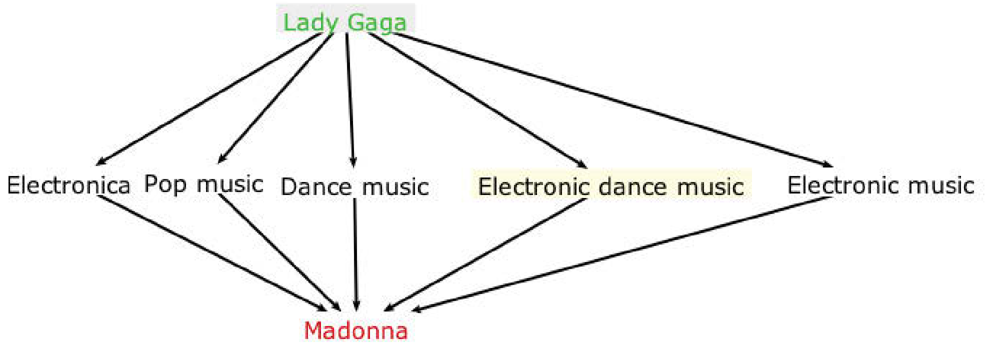


Figure 5.4: Path Length 2: Explanation of path based enrichments over the Artist-Genre edge set. The user can see the different nodes that were used for the enrichment with Madonna.

strategy is to weight edges based on their significance to connect a node in the data set. If the edge is the only one connecting a node, determined by the degree of a node, it is considered as more important than edges that connect a node with several other edges. For parallel edges/paths the ratings are summed up. For a sequence of edges the weights are multiplied and weighted by a discount factor (depending on the path length). In our system, we implemented the path-based approach using a breadth-first search algorithm with a limited search depth [179]. The search depth limit is set to make sure that the computed results are relevant for the input items and not only loosely connected. With the depth limit, no items are taken into account where the path length to the most relevant item is longer than the defined search limit.

Another advantage of path based approach is that no additional effort is needed for building a model. Thus, updates in the data set immediately affect the computed results.

Model-based predictions Real-world data sets are often sparse and noisy. In order to cope with these problems we reduce the complexity of the data set by aggregating similar entities into clusters. To assure that users still understand computed recommendations, we use Hierarchical Agglomerative Clustering [211] that combines entities with similar features in one cluster. The computed clusters are treated as nodes. Thus, path based search strate-

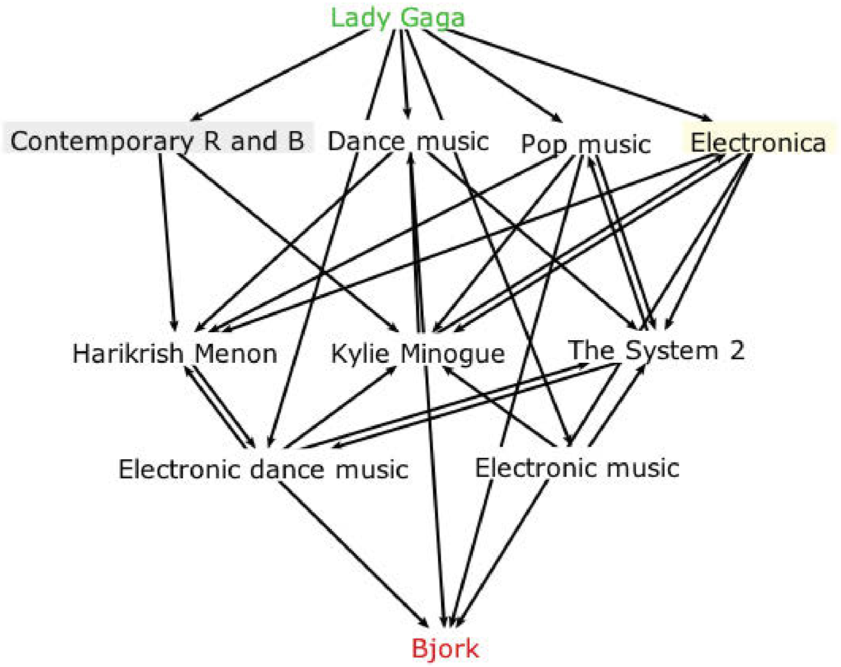


Figure 5.5: Path Length 4: Explanation of path based enrichments over the Artist-Genre edge set. The user can see the different nodes that were used for the enrichment with Björk.

gies can be used for searching relevant entities.

The advantages of model-based recommenders are that the complexity of the data set can be effectively reduced to speed-up the computation of relevant entities. Furthermore, the reduction of noise in large data sets often improves the result quality. The algorithm applied for reducing the graph complexity highly depends on the domain. We decided to focus on Hierarchical Agglomerative Clustering since it enabled us to choose similarity measures and clustering parameters optimized for each relationship set. Moreover, for recommendations computed based on clustered entity sets path based explanation can be provided. The disadvantages of model-based recommenders are that additional effort is needed for calculating and updating the model. A prediction based on clustering presented in Fig. 5.6. As the results of the cluster algorithm are most of the time only loosely related to the input node, the results from the clustering are not considered in the evaluation.

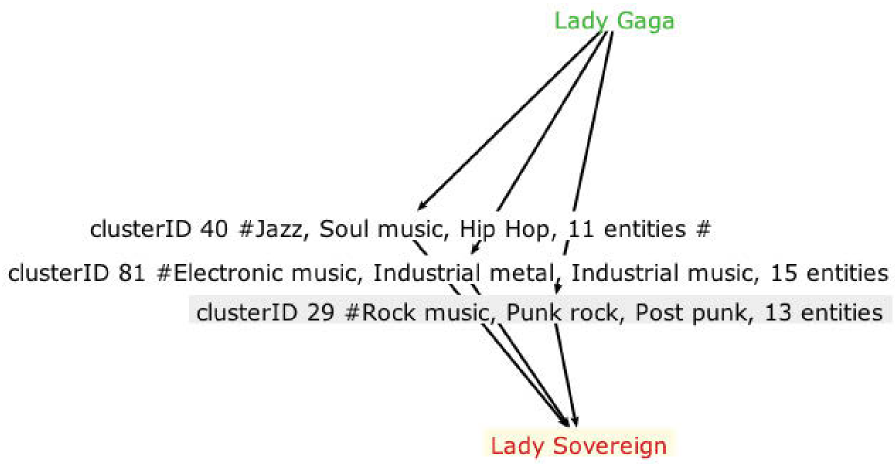

Lady Gaga
clusterID 40 #Jazz, Soul music, Hip Hop, 11 entities #
clusterID 81 #Electronic music, Industrial metal, Industrial music, 15 entities
clusterID 29 #Rock music, Punk rock, Post punk, 13 entities
Lady Sovereign

Figure 5.6: Cluster based prediction: Explanation of cluster based enrichments using automatically generated genre cluster.

5.3 Evaluation

The goal of the evaluation is to research the impact of an enriched user profile on the cold start problem for CF. We therefore consider two evaluation scenarios:

New user and new application: The first scenario covers the cold start problem for a new music recommendation application with few users. In this scenario, we analyze the effect of the enriched user profiles for a new music recommendation application that has a small number of users and how recommendation quality is affected for new users.

New user and big application: The second scenario is focused on a new user who joins a well established recommendation service, such as LastFM or Facebook. We study how the enrichment approach works for new users in a big recommendation application which already has a lot of users.

5.3.1 Data Sets

The evaluation is performed using two data sets from Facebook and LastFM collected between January and September 2010. We extracted data from around 60,000 users and kept the profiles that contain data about interests in music. For the evaluation we used all user profiles containing at least two music interests. Users from the Facebook data set expressed their interests by 'liking' an artist. Users in the LastFM data set showed their interests by listening to music, which is implicitly tracked information from LastFM, and by actively 'favoring' artists. The resulting Facebook data set consists of 3,011 users and 14,516 liked music items. The LastFM set consists of 7,743 users and 11,333 favored music items. We only crawled user profile information, no other data from Facebook, e.g. Facebook Open Graph[4] information, or data from LastFM about similar artists is part of the user profile data. The user profiles only contain the user name, the artist name or music album name, and in the LastFM set also the MusicBrainz ID[5].

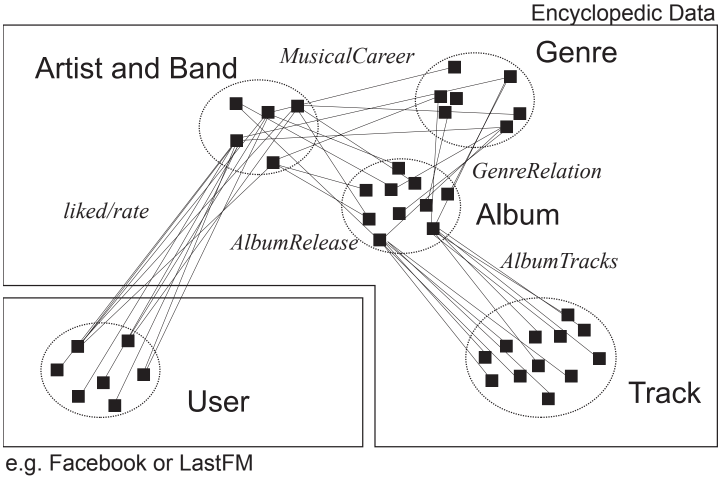


Figure 5.7: The semantic data set with music information and user profiles linked to it.

The semantic information that is needed for our approach is retrieved from Freebase. In our scenario, we make use of data from the music domain consisting of four music entity types, namely *Artists, Albums, Tracks*, and *Genres* relations between them. The relationship between artist and genres describes the genre in which an artists works; the relationship between album

[4] http://developers.facebook.com/docs/opengraph/
[5] http://musicbrainz.org/

and artists describes which artist can be found on an album release, and finally the relationship between album and genre defines a genre assignment for each album. The created data set is schematically visualized in Fig. 5.7. Table 5.1 shows the number of edges and entities contained in the data set.

Table 5.1: Music information contained in the Freebase data set.

Entities	number of entities	number of edges			
		Musicians	Genre	Albums	Tracks
Musicians	417217	–	79543	374445	–
Genre	3082	79543	–	90444	–
Albums	438180	37445	90444	–	1048565
Tracks	1048576	–	–	1048565	–

To analyze how semantic encyclopedic data can improve CF, we interlinked the semantic data set retrieved from Freebase with LastFM and Facebook as explained in Section 5.3.2.

5.3.2 Interlinking User Profiles

The extracted Facebook and LastFM profiles are initially isolated, meaning that there is no connection to the Freebase data set. However, our approach requires a graph containing the user profiles and the Freebase data interlinked. The linkage is needed as our enrichment algorithm is a graph-based method. Without connected data, the profile enrichment cannot be computed. Thus, it is necessary to know that an entity 'Facebook#The_Beatles' in a user profile is similar to the entity 'Freebase#Beatles' in the Freebase data set and to create a link between them. Fig. 5.8 shows the situation before and after the linkage. The linkage is done using a set of rules that connect the profiles. First, we check if we have a MusicBrainz ID (which is the case if we got the user data from LastFM). If we have the MusicBrainz ID the linkage is easy as this information is also part of the meta-information that Freebase provides about the artists. If no MusicBrainz ID is available we try to link entities based on the artist name in different spellings and languages offered by Freebase. If more than one Freebase node matches the rules and we cannot disambiguate the correct node this entity is disregarded. While we assume that this method minimize the number of false positive linked entities, there still may be incorrectly linked entities, which might lead to a reduced recommendation quality.

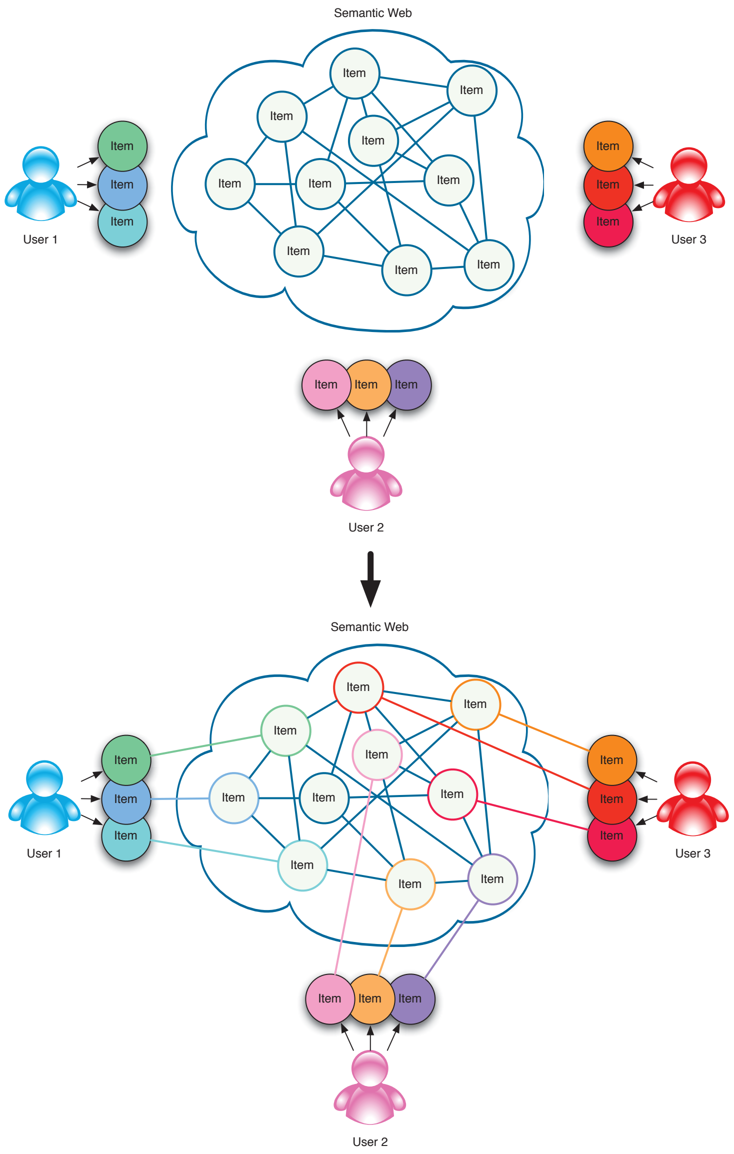
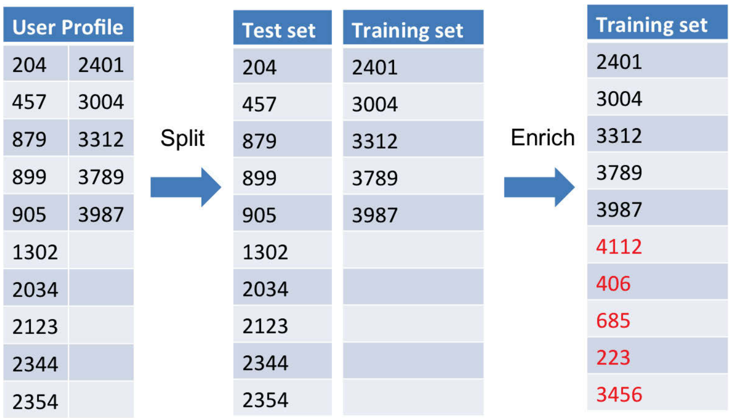


Figure 5.8: To compute the enriched profiles we first need to find edges between the user profiles and the semantic data set.

Having connected the user profiles with the Freebase data set, the derived semantic network can be used for enriching user profiles.

5.3.3 Selection of ‘New Users’ and evaluation algorithms

The selection of users who represent the new users joining a recommendation service was done by creating a subset with all users who have rated exactly 15 items. All test users must have the same number of items in the profile, to be able to compare the results of the different evaluation runs in the different scenarios. The number 15 was chosen because it gives a big enough user profile for the evaluation (train and test split) and also enough users with 15 items to have statistically enough test data. From these 15 items we use a set of ten items as our test set and for the training we arbitrarily choose one to five of a user’s remaining items for the initial user profile (training set) to simulate the cold start problem. The process is visualized in Fig. 5.9. We conduct several test runs, starting with a user profile containing only one item, and then iteratively increase the number of items up to five. The training set was enriched with additional five to nine items, depending on the initial size of the training set, so that it always contains ten items. Results are averaged over 200 evaluation runs for each user profile size (one to five items) using the following forms of CF algorithms:



User Profile	
204	2401
457	3004
879	3312
899	3789
905	3987
1302	
2034	
2123	
2344	
2354	

Test set	Training set
204	2401
457	3004
879	3312
899	3789
905	3987
1302	
2034	
2123	
2344	
2354	

Training set
2401
3004
3312
3789
3987
4112
406
685
223
3456

Figure 5.9: Example of a user profile with 15 item. First a training test split is done with ten test items and five items in the training set. Then we enrich the user profile with five additional items. This enriched set is then used for CF.

- **CF with standard profiles:** The Baseline. A standard CF algorithm using the Tanimoto coefficient [126] to compute the user similarity.

- **CF with enriched profiles:** The standard CF algorithm using the enriched user profiles instead of the standard profiles.

- **Most Popular Recommender:** A simple algorithm recommending the top n items of the data set.

- **CF + enriched profiles:** A combined method of the first two CF methods. If the standard CF does not find a recommendation, the CF with the enriched profiles is used. This approach avoids the recommendation depending mostly on the items used to enrich the profile.

- **CF + Most Popular recommendations:** An approach using most popular recommendations if the standard CF find no results.

- **Random Recommender:** Recommending randomly chosen items.

5.3.4 Evaluation of the 'New user and new application' scenario

The application is created by randomly selecting 5,000 users from our crawled data set. These users represent users who already using the application. The test users, which are different from the 5,000 users, are also chosen from the user data set containing only users with an interest in jazz or swing music. This was done to augment the cold start problem as most users in our data set have a "Pop" taste. The initial user profiles are enriched using the Freebase data. Fig. 5.10 presents the results for the different algorithms described in Section 5.3.3. The results show that the enrichment has a huge positive impact on the recommendation quality. Both approaches using the enriched profiles (CF using only enriched profiles and *CF with standard profiles* combined with *CF with enriched profiles*) clearly outperform the *standard CF* and *CF + Most Popular* for user profiles of size 1-4. For users with a user profile size of five, the *CF with enriched profiles* is slightly inferior than the *standard CF*. *Most Popular* and *Random* recommendation have no impact at all. Using only the *Most Popular* recommendations does not work as the selected test users were only interested in swing and jazz music as the common taste in the randomly selected data set is on pop music. Thus, the list of *Most Popular* recommendations consists of pop artist and does not contain any swing or jazz artists.

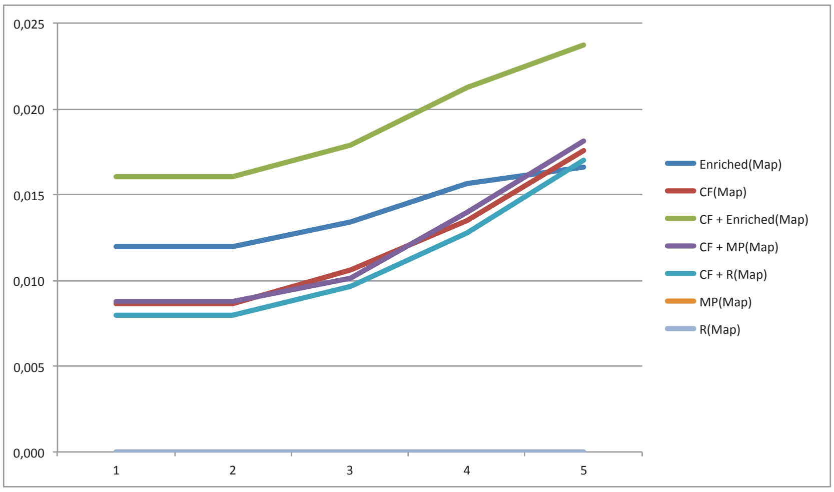


Figure 5.10: Cluster based prediction: Explanation of cluster based enrichments using automatically generated genre cluster.

Fig. 5.11 shows the percental change of recommendation quality compared to the *CF with standard profiles*. The usage of our enrichment approach improves the recommendation quality by over 90% for very small profiles (size one and two) and over 40% for the bigger profiles (size four and five).

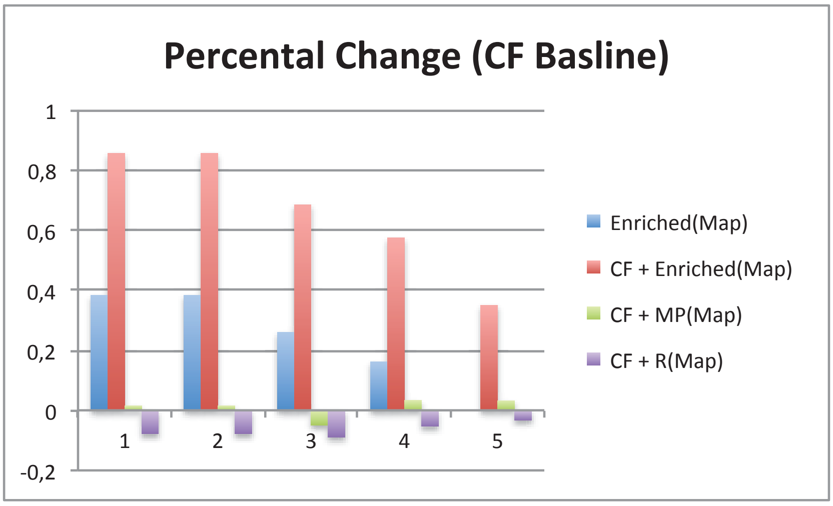


Figure 5.11: Cluster based prediction: Explanation of cluster based enrichments using automatically generated genre cluster.

5.3.5 Evaluation of complete Facebook and LastFM

The evaluation for the second scenario, the 'New user and big application' scenario, is done separately for Facebook and LastFM to compare if there are differences in a Social Network and a distinguished music recommendation service like LastFM. The evaluation covers three different user subsets:

1. Recommendations based on the complete data set.
2. Recommendations for users who have an uncommon taste. Which is similar to the swing and jazz user set used in the evaluation in Section 5.3.4.
3. Recommendations for users who mostly like popular artists.

The split between users with an unusual taste and users with a common popular taste is done based on the average deviation of popular artists in a user's profile. The popularity of an artist is computed based on the distribution in the Facebook and LastFM data sets. The initial user profiles (with 1 - 5 items) are enriched with five additional items from Freebase, so that the user profiles given to the collaborative filtering recommender have a size of six to ten items.

Fig. 5.12 and Fig. 5.13 show the evaluation results on both data sets. Results on the Facebook data set show that *CF with enriched user profiles* does not improve the recommendation quality compared to the *standard CF*. The enrichment even leads to a reduced precision.

This is expected, as our enrichment approach adds mostly 'popular' entities from Freebase to the user profile, meaning that the enrichment can blur the user profile and make the user profile less personal. As explained, the enrichment algorithm takes the degree of a node in the Freebase graph into account. Thus, mostly popular artist and genres are used for the enrichment. This is a shortcoming of the encyclopedic Freebase data set as there are no other indicators than distance to the user profile and degree of a node that could be used. Also the *standard CF* benefits from the fact that it is more likely to find similar users in a common taste scenario, hence recommended items bases on the original, not blurred, user profile and the CF can make use of more neighbors (similar users). On the other hand, a more detailed

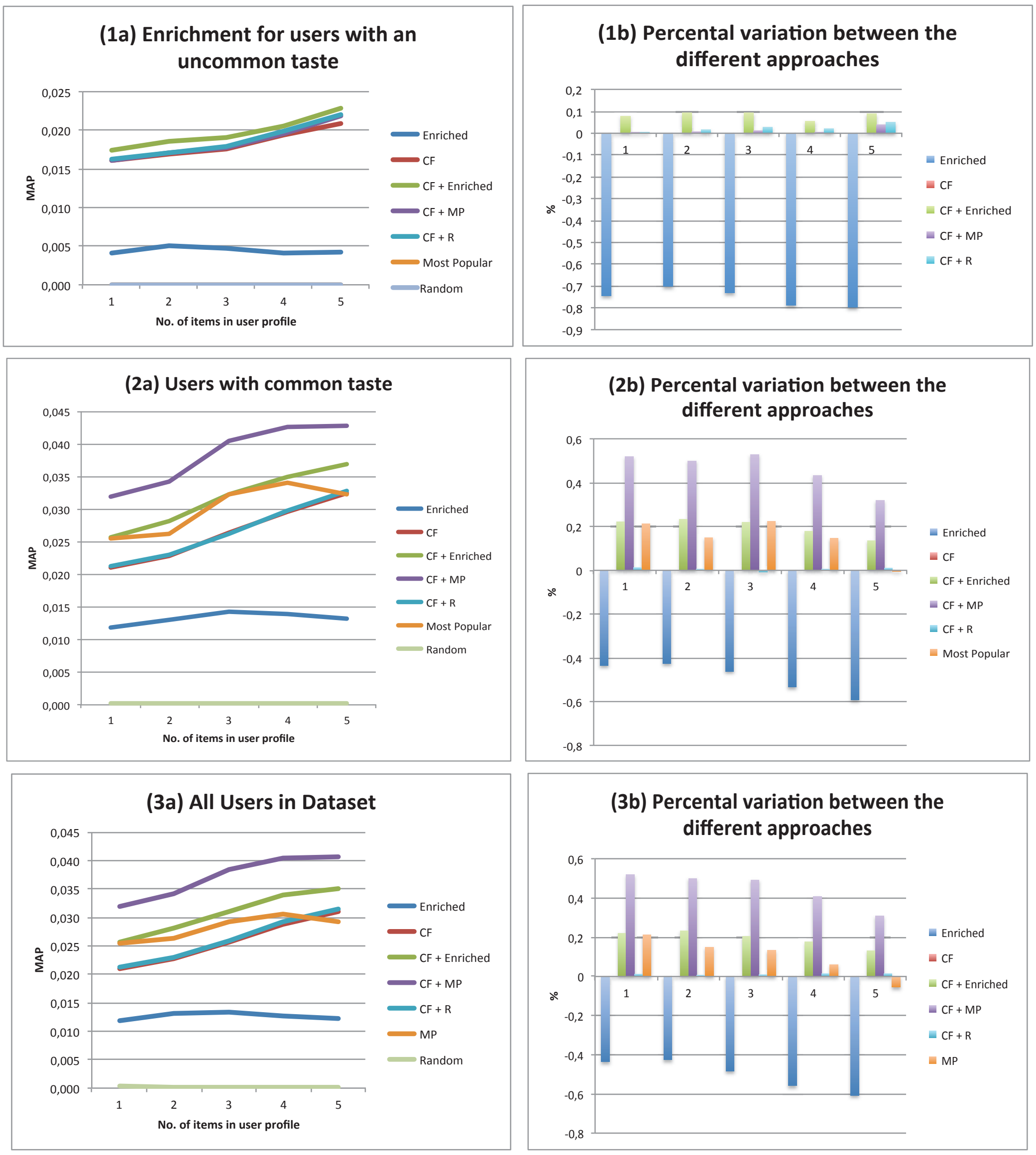


Figure 5.12: Results for Evaluation on the LastFM data set. (a) shows the results for users with an unusual music taste. (b) shows the results for users with popular music taste. (c) shows results over complete data set.

look on the results reveals that a combination of the standard and enriched CF can improve the quality. The reason is that in cases where *standard CF* cannot find appropriate items because no similar users can be found, the enrichment helps to find other users based on the enriched profile and hence items to recommend. This effect becomes even more visible in scenarios where recommendations for users with an uncommon taste are computed. In these scenarios the strategy *CF + enriched profiles* outperforms all other approaches. As CF depends on an sufficient amount of neighbors to compute recommendations and finding similar users for users with an uncommon taste is more difficult the enrichment helps to overcome this problem. The results for the LastFM user profiles confirm the findings on the Facebook data set. On the LastFM data set, we used a more restrict threshold to distinguish between users with common and uncommon taste. The evaluation results show that for users with a uncommon taste *CF + Most Popular* recommendations perform bad while the *CF + enriched profiles* recommender really improves recommendation quality. For common taste users and all users, the *CF + Most Popular* recommender performs best. Both combined strategies outperform the *standard CF* recommender.

5.4 Closing Discussion

In this chapter we presented a new semantic approach to overcome the cold start problem by enriching user profiles with data retrieved from semantic encyclopedic data sets. Our evaluation shows that, depending on the scenario, the profile enrichment improves the recommendation quality. Especially in scenarios where the given user profile is very small and the interests of a user differs from the mean taste of the other users (see Section 5.3.4). However, the evaluation also showed some shortcomings of the presented approach. The enrichment works very well for users with an unusual taste and in scenarios where the number of users of an application is low; in these scenarios the enriched profiles heighten the recommendation quality. In contrast, the enrichment is not helpful for users with large profiles or a popular music taste. In these cases the enrichment blurs the user profile and the therein-specified user taste, because the Freebase data contains general domain information, and for users with a common taste more or less universal knowledge is added. Adding the artist “Madonna” does not make sense for user profiles already containing a lot of pop artists; it only leads to more general profiles less tailored to the individual user preferences. Different strategies to overcome this problem are conceivable. On the one hand, our approach needs to weight the edge types in a more user centric way. A user may like an artist because of a certain song but does not like the complete discography; or a user might like the artist because of the social engagement of that artist and not be-

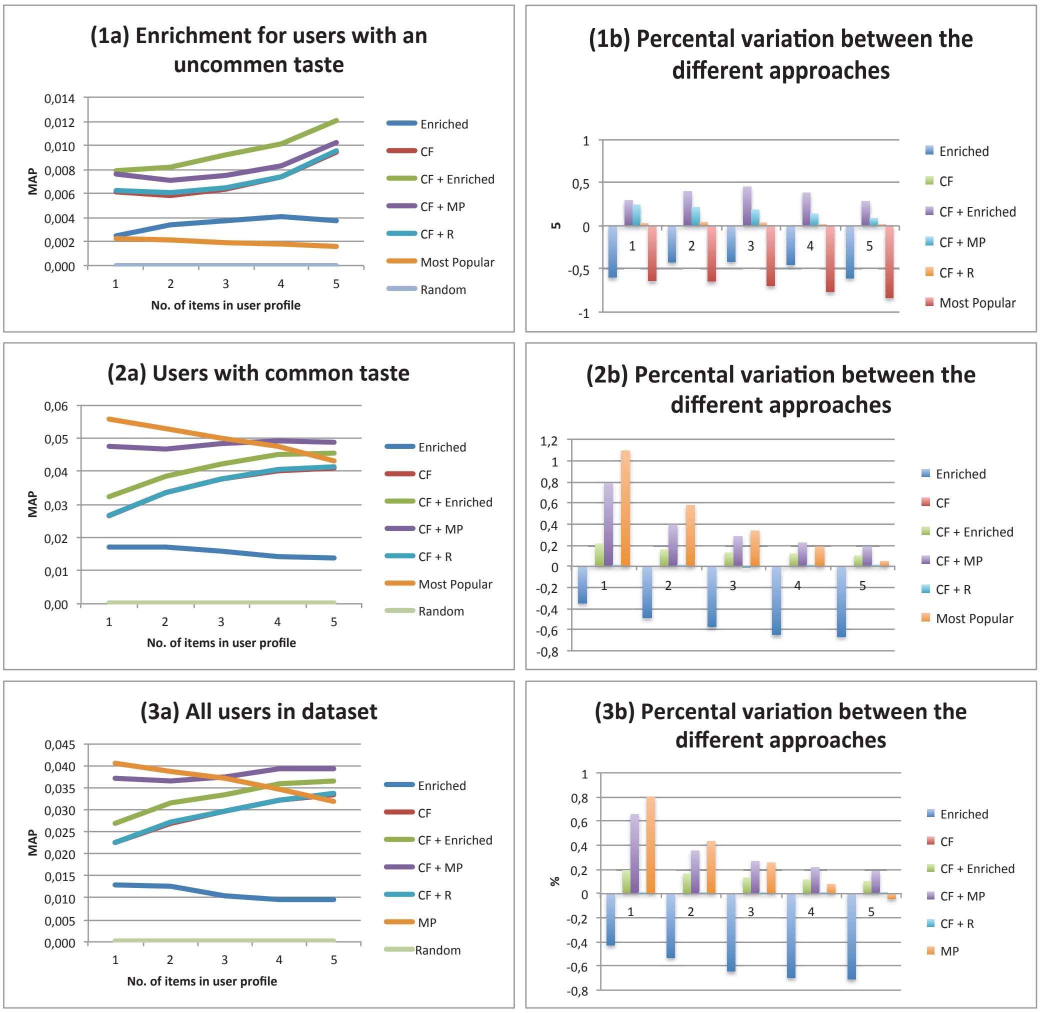


Figure 5.13: Results for Evaluation on the Facebook data set. (a) shows the results for users with an unusual music taste. (b) shows the results for users with popular music taste. (c) shows results over complete data set.

cause of the music. Therefore, more contextual information about users is needed, enabling a context sensitive weighting of the information used for the profile enrichment. The increasing popularity of Social Semantic Web approaches and standards like FOAF[6] can be one important step in this direction [42, 43]. On the other hand, semantic data sets itself have to be enriched with more meta-information about the data. General quality and significance information like prominence nodes and weighted relations can

[6] http://www.foaf-project.org/

improve semantic algorithms to better compute the importance of paths between nodes. An artist that made hundreds of bad albums may have a high number of links to e.g. a genre node, but is not an important artist for this genre while another artist made only one or two albums but defined a genre. In this case, a significant weight for the artists can improve the quality and performance of semantic algorithms.

The next step is to perform a live user evaluation asking them if the recommendation based on enriched profiles is of any help as offline evaluations hardly show the "real world" impact of such a user centric approach. Future steps are the evaluation of a focused enrichment, e.g. only using artist or genres information, based on the context of the user. Another direction is to implement a sophisticated weighting model (e.g., based on prominence, context and user groups) as an overlay for the Freebase data set, and to implement alternative network models (e.g., based on a low-rank approximation for the adjacency matrix of an relationship set [122]).

Chapter 6

Conclusion

This thesis discusses challenges related to the ever increasing amount of information in today's internet, namely the Social Web. To overcome this *Information Overload* problem, adaptation and personalization to user's needs has proven to be a successful approach [44]. The goal of this thesis is to present new ways to take advantage of the possibilities coming with the Semantic Web to unify and aggregate user information distributed in the Social Web with the goal to enhance adaptive systems. This thesis covers the different layers of an adaptive system and presents new approaches for the layers on how to take advantage of semantic technologies. In this process, semantic models and techniques are developed to capture user behavior and user information and aggregate this information to a user profile. This unified and aggregated user profile serves as an input for adaptive systems and allows for better personalization. In order to demonstrate the potentials for adaptive systems, the domain of recommender systems was chosen to show the benefits of a semantic models holds. The focus is placed on two problems in this domain - the *Cold Start Problem* and the *Grey Sheep Problem*.

In the following section, the contributions of this thesis are presented by revisiting the main research questions, which are identified in Section 1.2, and highlighting the results, effects and the scientific contributions of this thesis. Finally, further research opportunities are presented in the last section.

6.1 Research Questions Revisited

In Chapter 1, we introduced adaptive systems (see Chapter 1) and identified four tasks an adaptive system has to perform depicted in Fig. 6.1.

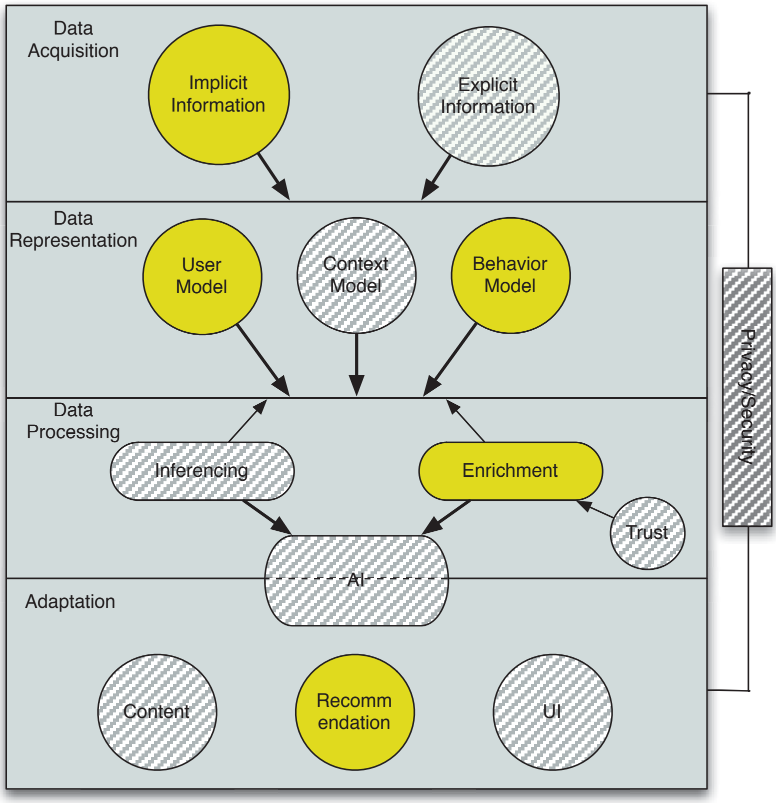


Figure 6.1: Parts of the Adaptive System that are processed in this work.

- The **data acquisition** task - collecting information about users,
- The **representation** and **data mining** task - processing information and build a user model,
- The **adaptation task** - applying the user model to adapt the application.

Along those tasks, we identified two thematic blocks where we placed our research questions.

Data Acquisition and Representation The first two research questions we defined and which align with the first and second layer of the adaptive architecture, are:

- *How can user behavior, and the meaning of a click be collected on today's highly dynamic websites?*
- *How can semantic user behavior be modeled?*

The first question is discussed in Section 3.2, which presents an in-depth examination of necessary functionality and solution approaches to collect more meaningful user behavior from websites. We discuss shortcomings of existing tracking approaches, that miss a lot of underlying semantics behind user actions, from which we derive requirements for a semantic tracking solution. Based on that, we present a **Semantic User Tracker** which shows how to take advantage of existing Semantic Web Technologies such as *RDFa* and *Microformats* to collect extra knowledge from user interactions on a website. The proposed semantic tracking solution is the first building block of a semantically enhanced adaptive system. The **User Behavior Ontology (UBO)** described in Section 3.3 provides a user model representation suitable for collecting and managing user behavior from complex websites. It follows the OWL standard, defined by the W3C, and serves as a general behavior model for tracking all user interactions, a semantic form of log files. While log files only capture explicit interaction, the presented tracking solution and the UBO now makes it possible to track, collect and manage also implicit interaction data. With the UBO, actual interaction is captured, in the *ubo:Event* class, as well as the site structure, using *ubo:View* and *ubo:Element*. The UBO allows, for instance, to collect a click on a link in a search result list, and additionally also to collect all other elements in that list including their order. With the semantic user behavior model, sharing and reusing of the collected information is possible. If a user shows interest in an artist like Madonna for instance, this information can be shared and reused by a music recommendation system. Using semantic technologies also allows using extra knowledge from external sources, such as DBpedia, to give more meaning to the collected data. The combination of the above described approaches, the tracking approach and the UBO is presented 'in action' in the **SERUM** system in Section 3.4. SERUM shows that the utilization of semantic technologies in a news recommendation system can enhance the usage experience by lowering the needed initial user information, thus tackling the *Cold Start Problem*.

With the presented UBO and the discussed semantic tracking method, we give answer to the research questions above. The tracking of user interaction can be improved by incorporating semantic technologies and with the UBO, the collected data can be utilized to improve recommendation quality, which is demonstrated in the SERUM showcase.

The next pair of questions is solely related to the second layer of the adaptive architecture and focuses on the semantic modeling and managing of user information from a Social Web perspective:

- *How must a semantic user model for the Social Web be constructed?*
- *How can we leverage Semantic Web technologies to aggregate user information from different web applications?*

We answer these questions in Chapter 4. In this chapter, we discuss and present two main approaches. We present a user model for the Social and Semantic Web, the **Semantic Web User Model - SWUM**, which covers all aspects a user model in that domain must cover. This is grounded by an extensive study of different social web applications where we identify information that needs to be taken into account for a Social Web user model. This is presented in Section 4.3. The SWUM is a generic model, based on OWL, that allows to manage, share and reuse user information. In order to be able to reuse such knowledge, strategies are needed to a) aggregate information from different applications and b) give information a semantic meaning. Therefore, we present in Section 4.2 a model to aggregate user information from different applications, the *Profile Data Model - PDM* and a system that uses this information to build and manage an aggregated user profile. This aggregated profile is enriched with additional information from the Semantic Web. We present a case study which shows the aggregation process and the usage of enriched profiles for personalized recommendations. The enrichment algorithm is explained in detail in Chapter 5. Also related to the second question, we present a **User Model Word Net** to store, manage and aggregate user information. This model is especially adapted to the needs of the Social Web and presented in Section 4.3.

The presented SWUM and the approach to aggregate user information from different applications answer above questions. The SWUM addresses the first question by providing a model for the Social Web, based on the presented study. The second question is addressed by the presented *User Model*

Word Net and the aggregation model *PDM*. The benefit of the presented approaches in helping to overcome the *Cold Start Problem*, is shown in the demo application 'My Personal User Interface', see Section 4.2.2.

Data Processing and Adaptation The last set of research questions we defined focus on the third and fourth layer of the adaptive system architecture presented in Fig. 6.1:

- *How can we leverage the growing knowledge in the Semantic Web to lower the initially needed amount of user preference data for Collaborative Filtering?*
- *How does a semantically enriched user profile influences recommendation quality?*

To answer those questions, we present a scenario of a user using a recommendation system. The user has only a few ratings so far, and we show how the presented semantic technologies can support the system to overcome the *Cold Start Problem* and the *Grey Sheep Problem.* We present an generic approach (see Section 5.2) that takes existing user profile information and tries to find related knowledge in the Semantic Web to enrich the user profile with additional information that helps to improve recommendation quality. We conducted a comprehensive evaluation of our approach, using two data sets, one is extracted from LastFM[1]; the other one is from Facebook[2], to see how the enriched user profiles affect CF recommendations. This is outlined in Chapter 5. We focus on the *Cold Start Problem* and the *Grey Sheep Problem.* The evaluation shows that taken into account data from the Semantic Web, initial recommendations can be improved.

6.2 Outlook on Further Research

The scope of this work was the extension of different parts of the adaptive system with semantic technologies. We therefore introduced different se-

[1] http://www.last.fm/

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

mantic buildings blocks for adaptive systems and presented the benefits that comes with them.

Out of scope for this thesis is the topic privacy and security of personal user information [208, 115]. Nevertheless, it is an important topic particularly with respect to the fact that the creation of possibilities to share and reuse data in different applications creates an extra need for users to be able to manage what is shared and how it is shared. The question to solve is what is the best personalization approach that considers privacy concerns and corresponding limitations. A brief discussion on how users can be enabled to manage what is shared is given in Section 4.2.2.

The continuation of the presented work should be carried out in the first and last layer of the adaptive system, the acquisition and adaption layer. Gathering information about the user is a crucial point for good personalization. While this work presented approaches to gather more implicit information in standard web usage and how to store and share it, a lot of future work can be done by using linked data to collect more related information [2]. In the adaptation layer multiple aspects have to be considered. The enrichment process has to be enhanced by incorporating more context and user information to select best matching semantic information for the given user model. Also the impact of the enriched user profiles for different algorithms, beside the here evaluated CF algorithms, has to be evaluated. And there, as always, room for improvements of the evaluated algorithms by parameter tweaking.

Follow up work on semantic user modeling includes a multilingual search assistant for the health-care domain [162, 161]. Goal of the work was providing migrants with a tool to find related information about the country's health-care system. Migrants can search for different health topics using their mother tongue, or mixed language queries, and retrieve results in their preferred language. Based on a health ontology and graph-based search algorithm, we deployed a semantic user model covering health data, demographic and location information. When a search query is submitted by a user, the system automatically extends the query by using user profile information. Searching for information about pregnancy, the system uses location information to present doctors, specialized for pregnancy, close by the user. A diabetes patient, searching for nutrition and sports information, gets results enclosing the diabetes information. While using semantically enhanced adaptive systems still requires some manual work, the multilingual search assistant shows that by using semantic user models, personalization can be enhanced.

Bibliography

[1] *Proceedings of the second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation 2011 (SPIM 2011)*, volume 781, Bonn, Germany, October 2011. CEUR-WS.org.

[2] Fabian Abel, Claudia Hauff, Geert-Jan Houben, and Ke Tao. Leveraging user modeling on the social web with linked data. In Marco Brambilla, Takehiro Tokuda, and Robert Tolksdorf, editors, *Web Engineering*, volume 7387 of *Lecture Notes in Computer Science*, pages 378–385. Springer Berlin / Heidelberg, 2012.

[3] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Interweaving public user profiles on the web. In Paul De Bra, Alfred Kobsa, and David N. Chin, editors, *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings*, volume 6075 of *Lecture Notes in Computer Science*, pages 16–27. Springer, 2010.

[4] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction (UMUAI), Special Issue on Personalization in Social Web Systems*, pages 1–42, 2011.

[5] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, UK, 1999. Springer Berlin / Heidelberg.

[6] Ben Adida, Ivan Herman, Manu Sporny, and Mark Birbeck. Rdfa 1.1 primer. Specification Document, June 2012.

[7] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alexander Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32(3):67–80, 2011.

[8] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Trans. on Knowledge and Data Engineering*, 17:734–749, 2005.

[9] James Frederick Allen. *A plan-based approach to speech act recognition.* PhD thesis, University of Toronto, Canada, 1979.

[10] Xavier Amatriain. Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.*, 14(2):37–48, December 2012.

[11] Sarabjot Singh Anand, Patricia Kearney, and Mary Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Trans. Internet Technol.*, 7:26, October 2007.

[12] Sarabjot Singh Anand and Bamshad Mobasher. Introduction to intelligent techniques for web personalization. *ACM Trans. Internet Technol.*, 7:4, October 2007.

[13] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207, 1995.

[14] Lora Aroyo, Peter Dolog, Geert-Jan Houben, Milos Kravcik, Ambjörn Naeve, Mikael Nilsson, and Fridolin Wild. Interoperability in personalized adaptive learning. *Educational Technology & Society*, 9(2):4–18, 2006.

[15] Ivon Arroyo and Beverly Park Woolf. Inferring learning and attitudes from a bayesian network of log file data. In *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 33–40, The Netherlands, 2005. IOS Press.

[16] Richard Atterer, Monika Wnuk, and Albrecht Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 203–212, USA, 2006. ACM.

[17] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, New York, 1999.

[18] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 805–810, USA, 2003. Morgan Kaufmann Publishers Inc.

[19] Ranieri Baraglia and Fabrizio Silvestri. Dynamic personalization of web sites without user intervention. *Commun. ACM*, 50:63–67, February 2007.

[20] David Beckett, Tim Berners-Lee, and Eric Prudh́ommeaux. Terse rdf triple language. Technical report, World Wide Web Consortium (W3C), August 2011.

[21] David Beckett and Brian McBride. Rdf/xml syntax specification (revised), February 2004.

[22] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Commun. ACM*, 35(12):29–38, 1992.

[23] Alejandro Bellogín, Iván Cantador, Fernando Díez, Pablo Castells, and Enrique Chavarriaga. An empirical comparison of social, collaborative filtering, and hybrid recommenders. *ACM Trans. Intell. Syst. Technol.*, 4(1):14:1–14:29, February 2013.

[24] David Benyon and Dianne Murray. Applying user modeling to human-computer interaction design. *Artificial Intelligence Review*, 7:199–225, 1993.

[25] Bettina Berendt. Using site semantics to analyze, visualize, and support navigation. *Data Mining and Knowledge Discovery*, 6:37–59, 2002. 10.1023/A:1013280719795.

[26] Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In Ian Horrocks and James Hendler, editors, *The Semantic Web ISWC 2002*, volume 2342 of *Lecture Notes in Computer Science*, pages 264–278. Springer Berlin / Heidelberg, 2002.

[27] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 18(3):245–286, 2008.

[28] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-representation mediation of user models. *User Modeling and User-Adapted Interaction*, 19(1-2):35–63, 2009.

[29] Tim Berners-Lee. Information management: A proposal. Web Document, 1989. http://www.w3.org/History/1989/proposal.html, Last visited: 21.09.2010.

[30] Tim Berners-Lee. Www at 15 years: looking forward. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 1–1, New York, NY, USA, 2005. ACM.

[31] Tim Berners-Lee. Long live the web: A call for continued open standards and neutrality, November 2010. http://www.scientificamerican.com/article.cfm?id=long-live-the-web&page=3, Last visited: March 2012.

[32] Tim Berners-Lee. An rdf language for the semantic web - notation 3 logic, September 2011.

[33] Tim Berners-Lee and Dan Connolly. Notation3 (n3): A readable rdf syntax. Technical report, World Wide Web Consortium (W3C), March 2011.

[34] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform resource identifier (uri) generic syntax, January 2005.

[35] Tim Berners-Lee, James Hendler, and Oro Lassila. The semantic web. *Scientific American*, 284:34–43, 2011.

[36] Mikhail Bilenko and Ryen W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 51–60, New York, NY, USA, 2008. ACM.

[37] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, February 2000.

[38] Pradipta Biswas, Peter Robinson, and Patrick Langdon. Designing inclusive interfaces through user modeling and simulation. *International Journal of Human-Computer Interaction*, 28(1):1–33, 2012.

[39] Chris Bizer, Sören Auer, Georgi Kobilarov, Jens Lehmann, Christian Becker, and Sebastian Hellmann. Dbpedia - querying wikipedia like a database and an interlinking-hub in the web of data, April 2009.

[40] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[41] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (ldow2008). In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1265–1266, New York, NY, USA, 2008. ACM.

[42] Uldis Bojars, Alexandre Passant, John G. Breslin, and Stefan Decker. Data portability with sioc and foaf. In *XTech*, 2008.

[43] Uldis Bojars, Alexandre Passant, John G. Breslin, and Stefan Decker. Social network and data portability using semantic web technologies. In *BIS 2008 Workshops Proceedings: Social Aspects of the Web (SAW 2008), Advances in Accessing Deep Web (ADW 2008), E-Learning for Business Needs*, volume 333 of *CEUR Workshop Proceedings*, pages 5–19. CEUR-WS.org, 2008.

[44] Paul De Bra and Wolfgang Nejdl, editors. *Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference, AH 2004, Eindhoven, The Netherlands, August 23-26, 2004, Proceedings*, volume 3137 of *Lecture Notes in Computer Science*. Springer, 2004.

[45] John Breslin and Stefan Decker. The future of social networks on the internet: The need for semantics. *Internet Computing, IEEE*, 11(6):86–90, 2007.

[46] John G. Breslin, Alexandre Passant, and Stefan Decker. *The Social Semantic Web*. Springer, Berlin, 2009.

[47] Dan Brickley and Libby Miller. Foaf vocabulary specification 0.91. Namespace Document, November 2007.

[48] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Model. User-Adapt. Interact.*, 6(2-3):87–129, 1996.

[49] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors. *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer-Verlag, Berlin Heidelberg New York, 2007.

[50] Peter Brusilovsky and Eva Millan. User models for adaptive hypermedia and adaptive educational systems. In Brusilovsky et al. [49], chapter 1, pages 3–53.

[51] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[52] Robin Burke. Hybrid web recommender systems. In Brusilovsky et al. [49], chapter 12, pages 377–408.

[53] Francesca Carmagnola and Federica Cena. User identification for cross-system personalisation. *Inf. Sci.*, 179(1-2):16–32, January 2009.

[54] Francesca Carmagnola and Federica Cena. User identification for cross-system personalisation. *Inf. Sci.*, 179(1-2):16–32, January 2009.

[55] Brian Carr and Ira P. Goldstein. Overlays - a theory of modeling for computer-aided instruction. Technical report, AI Lab Memo 406 MIT, 1977.

[56] Tantek Celik and Kevin Marks. rel=”tag”. Draft specification, Microformats.com, January 2005.

[57] Tantek Celik and Brian Suda. hcard 1.0. Specifiaction, Microformats.org, April 2010.

[58] Federica Cena, Antonina Dattolo, Ernesto William Luca, Pasquale Lops, Till Plumbaum, and Julita Vassileva. Semantic adaptive social web. In Liliana Ardissono and Tsvi Kuflik, editors, *Advances in User Modeling*, volume 7138 of *Lecture Notes in Computer Science*, pages 176–180. Springer Berlin / Heidelberg, 2012.

[59] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *CHI ’01 extended abstracts on Human factors in computing systems*, CHI EA ’01, pages 281–282, USA, 2001. ACM.

[60] Ed H. Chi, H. Chi, Adam Rosien, and Jeffrey Heer. Lumberjack: Intelligent discovery and analysis of web user traffic composition. In *In Proceedings of ACMSIGKDD Workshop on Web Mining for Usage Patterns and User Profiles*, pages 1–15. ACM Press, 2002.

[61] Philip R. Cohen and C. Raymond Perrault. Elements of a plan-based theory of speech acts. *Readings in natural language processing*, pages 423–440, 1986.

[62] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46(1):41 – 64, 2003.

[63] Honghua Dai and Bomshad Mobasher. Integrating semantic knowledge with web usage mining for personalization. In Anthony Scime, editor, *Web Mining: Applications and Techniques*. IRM Press, Idea Group Publishing, 2005.

[64] Ernesto William De Luca, Till Plumbaum, Jerome Kunegis, and Sahin Albayrak. Multilingual ontology-based user profile enrichment. In *1st Workshop on the Multilingual Semantic Web*, 2010.

[65] M. Dean and G. Schreiber. Owl, web ontology language. W3c recommendation, World Wide Web Consortium (W3C), February 2004.

[66] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer US, 2011.

[67] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5:4–7, 2001.

[68] Rezarta Islamaj Dogan, G. Craig Murray, Aurélie Névéol, and Zhiyong Lu. Understanding pubmed user search behavior through log analysis. online, November 2009.

[69] Paul Dourish. What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8:19–30, February 2004.

[70] Susan Dumais, Robin Jeffries, DanielM. Russell, Diane Tang, and Jaime Teevan. Understanding user behavior through log data and analysis. In Judith S. Olson and Wendy A. Kellogg, editors, *Ways of Knowing in HCI*, pages 349–372. Springer US, 2014.

[71] Economist. Break down these walls. The Economist, 2008. Last visitied: March 2012.

[72] Carsten Eickhoff, Jaime Teevan, Ryen White, and Susan Dumais. Lessons from the journey: A query log analysis of within-session learning. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 223–232, USA, 2014. ACM.

[73] Federico Michele Facca and Pier Luca Lanzi. Mining interesting knowledge from weblogs: a survey. *Data & Knowledge Engineering*, 53(3):225 – 241, 2005.

[74] Alexander Felfernig and Robin Burke. Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce*, ICEC '08, pages 3:1–3:10, USA, 2008. ACM.

[75] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. Developing constraint-based recommenders. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 187–215. Springer US, 2011.

[76] Kurt D. Fenstermacher and Mark Ginsburg. Mining client-side activity for personalization. In *Advanced Issues of E-Commerce and Web-Based Information Systems, 2002. (WECWIS 2002). Proceedings. Fourth IEEE International Workshop on*, pages 205–212, 2002.

[77] Tim W. Finin. Gums: A general user modeling shell. In Alfred Kobsa and Wolfgang Wahlster, editors, *User Models in Dialog Systems*, pages 411–430. Springer Berlin / Heidelberg, 1989.

[78] Josef Fink. *User Modeling Servers Requirements, Design, and Evaluation*. PhD thesis, Universität Duisburg-Essen, 2003.

[79] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 231–240, New York, NY, USA, 2007. ACM.

[80] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. In Brusilovsky et al. [49], chapter 2, pages 54–89.

[81] Neha Goel and C. K. Jha. Article: Analyzing users behavior from web access logs using automated log analyzer tool. *International Journal*

of Computer Applications, 62(2):29–33, January 2013. Published by Foundation of Computer Science, New York, USA.

[82] Ira P. Goldstein. The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11(1):51 – 77, 1979.

[83] Carrie Grimes, Diane Tang, and Daniel M. Russell. Query Logs Alone are not Enough. In Einat Amitay, Craig G. Murray, and Jaime Teevan, editors, *Query Log Analysis: Social And Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007)*, May 2007.

[84] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, 1995.

[85] Michael Grüninger and Mark S. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[86] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum.-Comput. Stud.*, 43(5-6):625–640, 1995.

[87] Eran Hammer-Lahav. The oauth 1.0 protocol. Request for comments. Internet Engineering Task Force, April 2010.

[88] Dominik Heckmann. *Ubiquitous User Modeling*. Akademische Verlagsgesellschaft Aka GmbH, Berlin, 2006.

[89] Dominik Heckmann, Tim Schwartz, Boris Brandherm, and Alexander Kröner. Decentralized user modeling with userml and gumo. In *Proceedings of the Workshop on Decentralized, Agent Based and Social Approaches to User Modelling (DASUM 2005)*, pages 61–65, 2005.

[90] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. Gumo - the general user model ontology. In *User Modeling*, pages 428–432, 2005.

[91] Dominik Heckmann, Eric Schwarzkopf, Junichiro Mori, Dietmar Dengler, and Alexander Kröner. The user model and context ontology gumo revisited for future web 2.0 extensions. In Paolo Bouquet, Jrme Euzenat, Chiara Ghidini, Deborah L. McGuinness, Luciano Serafini,

Pavel Shvaiko, and Holger Wache, editors, *Proceedings of the International Workshop on Contexts and Ontologies: Representation and Reasoning*, volume 298 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[92] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, 1999.

[93] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[94] Manuel Hernando. Student procedural knowledge inference through item response theory. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization*, UMAP'11, pages 426–429, Berlin, Heidelberg, 2011. Springer-Verlag.

[95] Erland Hjelmquist, Ulf Dahlstrand, and Lisbeth Hedelin. Visually impaired persons' comprehension of text presented with speech synthesis. *Journal of Visual Impairment & Blindness*, 86(10):426–428, 1992.

[96] Erland Hjelmquist, Bengt Jansson, and Gunnar Torell. Computer-orinted technology for blind readers. *Journal of Visual Impairment and Blindness*, 17:210–215, 1990.

[97] Tanja Hölldobler and Stefan Michel. Personalized shopping in the web by monitoring the customer. In *Proceedings of The Active Web*, 1999.

[98] Frank Hopfgartner, editor. *Smart Information Systems: Computational Intelligence for Real-Life Applications.* Advances in Computer Vision and Pattern Recognition. Springer, 2015.

[99] Gardner Howard. *Frames of Mind: The Theory of Multiple Intelligences.* Basic Books, New York, 1983.

[100] T. Hussain, S. Asghar, and N. Masood. Web usage mining: A survey on preprocessing of web log file. In *Information and Emerging Technologies (ICIET), 2010 International Conference on*, pages 1–6, June 2010.

[101] Isto Huvila. Where does the information come from? information source use patterns in wikipedia. *Information Research*, 15(3), September 2010.

[102] Jacob Jacoby. Perspecitves on information overload. *Journal of Consumer Research*, 10(4):432–436, 1984.

[103] Anthony Jameson. Modelling both the context and the user. *Personal and Ubiquitous Computing*, 5:29–33, 2001.

[104] Anthony Jameson. Adaptive interfaces and agents. In Andrew Sears and Julie A. Jacko, editors, *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, pages 433–458. CRC Press, USA, 2nd edition, 2008.

[105] Pavan Kapanipathi, Fabrizio Orlandi, Amit Sheth, and Alexandre Passant. Personalized filtering of the twitter stream. In *2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation*, 2011.

[106] Judy Kay. Stereotypes, student models and scrutability. In Gilles Gauthier, Claude Frasson, and Kurt VanLehn, editors, *Intelligent Tutoring Systems*, volume 1839 of *Lecture Notes in Computer Science*, pages 19–30. Springer Berlin / Heidelberg, 2000.

[107] Judy Kay, Bob Kummerfeld, and Piers Lauder. Personis: A server for user models. In Paul De Bra, Peter Brusilovsky, and Ricardo Conejo, editors, *AH*, volume 2347 of *Lecture Notes in Computer Science*, pages 203–212. Springer, 2002.

[108] Garrett Serack Keith Ballinger, Bill Barnes and James Causey. Patterns for supporting information cards at web sites: Personal cards for sign-up and sign-in. Technical report, Microsoft, 2007.

[109] Declan Kelly and Brendan Tangney. Using multiple intelligence informed resources in an adaptive system. In Mitsuru Ikeda, Kevin Ashley, and Tak-Wai Chan, editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 412–421. Springer Berlin / Heidelberg, 2006.

[110] G. Klyne and J. J. Carroll. Resource description framework (rdf): Concepts and abstract syntax, 2004.

[111] Stefan Werner Knoll, Till Plumbaum, Jan Leif Hoffmann, and Ernesto Wiliam De Luca. Collaboration ontology: Applying collaboration knowledge to a generic group support system. In *Group Decision and Negotiation Meeting*, pages 12–26, 2010.

[112] Stefan Werner Knoll, Till Plumbaum, and Ernesto Wiliam De Luca. Semantic group support system for context adaptive collaboration. In *CHI 2010 workshop on Context-Adaptive Interaction for Collaborative Work (CAICOLL 2010)*, 2010.

[113] Alfred Kobsa. User modeling: Recent work, prospects and hazards. In U. Malinowski M. Schneider-Hufschmidt, T. Kühme, editor, *Adaptive User Interfaces: Principles and Practice*, pages 111–128. Amsterdam: North-Holland, 1993.

[114] Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11:49–63, March 2001.

[115] Alfred Kobsa. Privacy-enhanced personalization. *Commun. ACM*, 50(8):24–33, August 2007.

[116] Alfred Kobsa, Jürgen Koenemann, and Wolfgang Pohl. Personalised hypermedia presentation techniques for improving online customer relationships. *Knowl. Eng. Rev.*, 16:111–155, March 2001.

[117] Nora Koch. *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. PhD thesis, Ludwig-Maximilians-University Munich, 2001.

[118] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, John Riedl, and High Volume. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40:77–87, 1997.

[119] Alexander Korth, Benjamin Hirsch, Till Plumbaum, and Andreas Nürnberger. A trilogy of webs for machines. In *Proceedings of the Workshop on Linked Data in the Future Internet at the Future Internet Assembly (LinkedDataFIA)*, December 2010.

[120] Alexander Korth and Till Plumbaum. A framework for ubiquitous user modeling. In *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on Information Reuse and Integration*, pages 291–297, August 2007.

[121] Tsvi Kuflik. Semantically-enhanced user models mediation: Research agenda. In *5th international workshop on ubiquitous user modeling (UbiqUM 2008)*, January 2008.

[122] Jérôme Kunegis and Andreas Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1–8, New York, NY, USA, 2009. ACM.

[123] Erwin Leonardi, Fabian Abel, Dominikus Heckmann, Eelco Herder, Jan Hidders, and Geert-Jan Houben. A flexible rule-based method for interlinking, integrating, and enriching user data. In Boualem Benatallah, Fabio Casati, Gerti Kappel, and Gustavo Rossi, editors, *Proceedings of the 10th International Conference on Web Engineering*, volume 6189 of *Lecture Notes in Computer Science*, pages 322–337, Vienna, July 2010. Springer Verlag.

[124] Kurt Lewin. *Principles of Topological Psychology.* Magraw-Hill Book Company, Inc., 1936.

[125] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76 – 80, 2003.

[126] Alan H Lipkus. A proof of the triangle inequality for the tanimoto distance. *Journal of Mathematical Chemistry*, 26(1-3):263–265, 1999.

[127] Bing Liu, Bamshad Mobasher, and Olfa Nasraoui. Web usage mining. In *Web Data Mining*, pages 527–603. Springer Berlin / Heidelberg, 2011.

[128] Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak. An architecture for smart semantic recommender applications. In *11th International Conference on Innovative Internet Community Systems*, pages 105–114, Berlin, 2011. LNI volume: P-186.

[129] Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak. A linked dataverse knows better: Boosting recommendation quality using semantic knowledge. In *Proc. of the 5th Intl. Conf. on Advances in Semantic Processing*, pages 97 – 103, USA, 2011. IARIA.

[130] Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

[131] Naresh K. Malhotra. Reflections on the information overload paradigm in consumer decision making. *Journal of Consumer Research*, 10(4):436–440, 1984.

[132] Frank Manola and Eric Miller. *RDF Primer*. W3C Recommendation. World Wide Web Consortium, February 2004.

[133] Abraham Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, 1943.

[134] Deborah L McGuinnes and Frank van Harmelen. Owl web ontology language overview. Technical report, World Wide Web Consortium (W3C), February 2004.

[135] Michael Meder, Till Plumbaum, and Frank Hopfgartner. Daiknow: A gamified enterprise bookmarking system. In Maarten Rijke, Tom Kenter, ArjenP. Vries, ChengXiang Zhai, Franciska Jong, Kira Radinsky, and Katja Hofmann, editors, *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, pages 759–762. Springer International Publishing, 2014.

[136] Bhaskar Mehta. *Cross system personalization: enabling personalization across multiple systems:*. PhD thesis, Universität Duisburg-Essen, 2008.

[137] Bhaskar Mehta, Claudia Niederée, Avare Stewart, Marco Degemmis, Pasquale Lops, and Giovanni Semeraro. Ontologically-enriched unified user modeling for cross-system personalization. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 119–123. Springer Berlin / Heidelberg, 2005.

[138] Florian Metze, Christian Bauckhage, and Tansu Alpcan. The ”spree” expert finding system. In *Proceedings of the First IEEE International Conference on Semantic Computing*, pages 551–558. IEEE Computer Society, 2007.

[139] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22:54–88, January 2004.

[140] Bamshad Mobasher. Data mining for web personalization. In Brusilovsky et al. [49], chapter 3, pages 90–135.

[141] Florian Mueller and Andrea Lockerd. Cheese: tracking mouse movement activity on websites, a tool for user modeling. In *CHI '01 extended abstracts on Human factors in computing systems*, CHI EA '01, pages 279–280, New York, NY, USA, 2001. ACM.

[142] Kim Anh Pham Ngoc, Young-Koo Lee, and Sung-Young Lee. Owl-based user preference and behavior routine ontology for ubiquitous system. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3761 of *Lecture Notes in Computer Science*, pages 1615–1622. Springer Berlin / Heidelberg, 2005.

[143] Claudia Niederée, Avare Stewart, Bhaskar Mehta, and Matthias Hemmje. A multi-dimensional, unified user model for cross-system personalization. In Liliana Ardissono and Giovanni Semeraro, editors, *Proceedings of the AVI Workshop on Environments for Personalized Information Access*, pages 34–54, 2004.

[144] Tadashi Nomoto. Re-ranking bibliographic records for personalized library search. In *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, JCDL '12, pages 125–128, New York, NY, USA, 2012. ACM.

[145] Natalya F. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Rec.*, 33:65–70, December 2004.

[146] Tim O'Reilly. O'reilly network: What is web 2.0, September 2005.

[147] Katherine H. Packer and Dagobert Soergel. The importance of sdi for current awareness in fields with severe scatter of information. *Journal of the American Society for Information Science*, 30(3):125–135, 1979.

[148] Balaji Padmanabhan, Zhiqiang Zheng, and Steven O. Kimbrough. Personalization from incomplete data: What you don't know can hurt. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 154–163, New York, NY, USA, 2001. ACM.

[149] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[150] Helena Sofia Pinto and João P. Martins. Ontologies: How can they be built? *Knowledge and Information Systems*, 6(4):441–464, July 2004.

[151] Danuta Ploch, Leonhard Hennig, Angelina Duka, Ernesto William De Luca, and Sahin Albayrak. Gerned: A german corpus for named entity disambiguation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LRECÍ2)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

[152] Till Plumbaum. Semantically-enhanced ubiquitous user modeling. In Paul De Bra, Alfred Kobsa, and David N. Chin, editors, *UMAP*, volume 6075 of *Lecture Notes in Computer Science*. Springer, 2010.

[153] Till Plumbaum. User behavior ontology, http://ubo-ontology.org. Webpage, 2011.

[154] Till Plumbaum. Semantic web user model ontology, http://swum-ontology.org. Webpage, 2012.

[155] Till Plumbaum and Benjamin Kille. Personalized fashion advice. In Hopfgartner [98], pages 213–237.

[156] Till Plumbaum, Funda Klein-Ellinghaus, Anna Reeske, Kristin Pelz, and Frank Hopfgartner. Health assistance for immigrants. In Hopfgartner [98], pages 75–97.

[157] Till Plumbaum and Andreas Lommatzsch. Personalized information access using semantic knowledge. In Hopfgartner [98], pages 181–211.

[158] Till Plumbaum, Andreas Lommatzsch, Ernesto William Luca, and Sahin Albayrak. Serum: Collecting semantic user behavior for improved news recommendations. In Liliana Ardissono and Tsvi Kuflik, editors, *Advances in User Modeling*, volume 7138 of *Lecture Notes in Computer Science*, pages 402–405. Springer Berlin Heidelberg, 2012.

[159] Till Plumbaum, Andreas Lommatzsch, Ernesto William De Luca, and Sahin Albayrak. Serum: Collecting semantic user behavior for improved news recommendations. In *UMAP 2011, Poster and Demo Session*, Spain, 2011.

[160] Till Plumbaum, Andreas Lommatzsch, Stefan Rudnitzki, Ernesto William De Luca, Holger Düwiger, and Sahin Albayrak. Adaptive music news recommendations based on large semantic datasets. In *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.

[161] Till Plumbaum, Sascha Narr, Elif Eryilmaz, Frank Hopfgartner, Funda Klein-Ellinghaus, Anna Reeske, and Sahin Albayrak. Providing multilingual access to health-related content. In *Proceedings of the 25th European Medical Informatics Conference*, 2014.

[162] Till Plumbaum, Sascha Narr, Veit Schwartze, Frank Hopfgartner, and Sahin Albayrak. An intelligent health assistant for migrants. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, 2013.

[163] Till Plumbaum, Katja Schulz, Martin Kurze, and Sahin Albayrak. My personal user interface: A semantic user-centric approach to manage and share user information. In *HCI International 2011*, 2011.

[164] Till Plumbaum, Katja Schulz, and Tino Stelter. Verbessertes profilmanagementsystem, 2011. Patent.

[165] Till Plumbaum, Tino Stelter, and Alexander Korth. Semantic web usage mining: Using semantics to understand user intentions. In *UMAP '09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization*, pages 391–396, Germany, 2009. Springer-Verlag.

[166] Till Plumbaum, Songxuan Wu, Ernesto William De Luca, and Sahin Albayrak. User modeling for the social semantic web. In *Proceedings of the second Workshop on Semantic Personalized Information Management: Retrieval and Recommendation* [1], pages 78–89.

[167] Axel Pols. Daten zur informationsgesellschaft. Technical report, Bitkom, 2007.

[168] Alexandrin Popescul and Lyle H. Ungar. Statistical relational learning for link prediction. In *Proceedings of the Workshop on Learning Statistical Models from Relational Data*, 2003.

[169] Eric Prudhommeaux and A. Seaborne. Sparql query language for rdf, January 2008.

[170] Liana Razmerita. Modeling behavior of users in adaptive and semantic enhanced information systems: The role of a user ontology. In *Adaptive Hypermedia and Adaptive Web-Based Systems 2008*, Hannover, Germany, August 2008. Springer Berlin / Heidelberg.

[171] Liana Razmerita. An ontology-based framework for modeling user behavior - a case study in knowledge management. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(4):772–783, July 2011.

[172] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, DIM '06, pages 11–16, USA, 2006. ACM.

[173] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In Ricci et al. [174], pages 1–35.

[174] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[175] Elaine Rich. User modeling via stereotypes. *Cognitive Science*, 3(4):329–354, 1979.

[176] Elaine Rich. Users are individuals: individualizing user models. *International Journal of Man-Machine Studies*, 18:199–214, 1983.

[177] Richard Riding and Stephen Rayner. *Cognitive styles and learning strategies: Understanding style differences in learning and behaviour*. David Fulton Publishers Ltd, London, 1998.

[178] Giuseppe Riva. Ambient intelligence in health care. *CyberPsychology & Behavior*, 6(3):296–300, 2003.

[179] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[180] Sharhida Zawani Saad and Udo Kruschwitz. Applying web usage mining for adaptive intranet navigation. In Allan Hanbury, Andreas Rauber, and Arjen P. Vries, editors, *Multidisciplinary Information Retrieval*, volume 6653 of *Lecture Notes in Computer Science*, pages 118–133. Springer Berlin Heidelberg, 2011.

[181] Alan Said, Brijnesh J. Jain, Sascha Narr, Till Plumbaum, Sahin Albayrak, and Christian Scheel. Estimating the magic barrier of recommender systems: a user study. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 1061–1062, New York, NY, USA, 2012. ACM.

[182] Suleyman Salin and Pinar Senkul. Using semantic information for web usage mining based recommendation. In *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*, pages 236–241, Sept 2009.

[183] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[184] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, pages 85–90, Washington, DC, USA, 1994. IEEE Computer Society.

[185] Kay-Uwe Schmidt, Ljiljana Stojanovic, Nenad Stojanovic, and Susan Thomas. On enriching ajax with semantics: The web personalization use case. In *4th European Semantic Web Conference*, June 2007.

[186] Matthias Schneider-Hufschmidt, Thomas Khme, and Uwe Malinowski, editors. *Adaptive User Interfaces: Principles and Practice (Human Factors in Information Technology)*. North Holland, 1993.

[187] Barry Schwartz. *The Paradox of Choice*. HarperCollins, 2009.

[188] Edwin J.(Ted) Selker. Cognitive adaptive computer help (coach): A case study. In Marvin V. Zelkowitz, editor, *Advances in Computers*, volume 47 of *Advances in Computers*, pages 67 – 140. Elsevier, 1998.

[189] Cyrus Shahabi and Farnoush Banaei-Kashani. A framework for efficient and anonymous web usage mining based on client-side tracking. In Ron Kohavi, BrijM. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 Mining Web Log Data Across All Customers Touch Points*, volume 2356 of *Lecture Notes in Computer Science*, pages 113–144. Springer Berlin / Heidelberg, 2002.

[190] Cyrus Shahabi, Amir M. Zarkesh, Jafar Adibi, and Vishal Shah. Knowledge discovery from users web-page navigation. In *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97) High Performance Database Management for Large-Scale Applications*, RIDE '97, pages 20–29, Washington, DC, USA, 1997. IEEE Computer Society.

[191] Derek Sleeman. Umfe: a user modelling front-end subsystem. *Int. J. Man-Mach. Stud.*, 23:71–88, July 1985.

[192] Sergey Sosnovsky and Darina Dicheva. Ontological technologies for user modelling. *Int. J. Metadata Semant. Ontologies*, 5(1):32–71, 2010.

[193] Richard A. Spreng and Robert D. Mackoy. An empirical examination of a model of perceived service quality and satisfaction. *Journal of Retailing*, 72(2):201–214, 1996.

[194] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, 2000.

[195] Kaye Stacey, Liz Sonenberg, Ann Nicholson, Tal Boneh, and Vicki Steinle. A teaching model exploiting cognitive conflict driven by a bayesian network. In Peter Brusilovsky, Albert Corbett, and Fiorella de Rosis, editors, *User Modeling 2003*, volume 2702 of *Lecture Notes in Computer Science*, pages 145–145. Springer Berlin / Heidelberg, 2003.

[196] Tino Stelter. Entwicklung eines user tracking systems zur verbesserten benutzermodellierung. Master's thesis, Technische Universität Berlin, 2008.

[197] Richard L. Street. Gender differences in health care provider-patient communication: are they due to style, stereotypes, or accommodation? *Patient Education and Counseling*, 48(3):201 – 206, 2002.

[198] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2, January 2009.

[199] Martin Szomszor, Harith Alani, Ivan Cantador, Kieron O'Hara, and Nigel Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In *Proceedings of the 7th International Conference on The Semantic Web*, ISWC '08, pages 632–648, Berlin, Heidelberg, 2008. Springer-Verlag.

[200] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Proceedings of Neural Information Processing Systems*, 2004.

[201] Jaime Teevan. The dangers of sharing log data. Online, May 2014. http://slowsearching.blogspot.de/2014/05/the-dangers-of-sharing-log-data.html, Last Visited: June 2014.

[202] Claudio Teixeira, Joaquim Sousa Pinto, Joaquim, and Arnaldo Martins. User profiles in organizational environments. *Campus-Wide Information Systems*, 25 Iss: 3:128–144, 2008.

[203] John Tolle. Transactional log analysis: Online catalogs. In Jennifer J. Kuehn, editor, *Research and Development in Information Retrieval, Sixth Annual International ACM SIGIR Conference, National Library of Medicine*, pages 147–160. ACM, 1983.

[204] Ilaria Torre. Adaptive systems in the era of the semantic and social web, a survey. *User Modeling and User-Adapted Interaction*, 19:433–486, December 2009.

[205] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136, 1996.

[206] K. van der Sluijs and G.-J. Houben. A generic component for exchanging user models between web-based systems. *Int. J. Continuing Education and Liflong Learning*, Vol. 16 Nos. 1/2:64–76, 2006.

[207] William van Winkle. Information overload. Website. http://www.gdrc.org/icts/i-overload/infoload.html, Last visited: February, 20th, 2014.

[208] Yang Wang and Alfred Kobsa. A pla-based privacy-enhancing user modeling framework and its evaluation. *User Modeling and User-Adapted Interaction*, 23(1):41–82, 2013.

[209] Daniel S Weld, Corin Anderson, Pedro Domingos, Oren Etzioni, Krzysztof Gajos, Tessa Lau, and Steve Wolfman. Automatically personalizing user interfaces. In *IJCAI*, volume 3, pages 1613–1619, 2003.

[210] Kung-Lu Wu, Philip S. Yu, and Allen Ballman. Speedtracer: A web usage mining and analysis tool. *IBM Systems Journal*, 37(1):89 – 105, 1998.

[211] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 515–524, USA, 2002. ACM.