Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.)

The Social Semantic Web 2007

Proceedings of the 1st Conference on Social Semantic Web (CSSW)

September 26-28, Leipzig, Germany

Gesellschaft für Informatik 2007

Lecture Notes in Informatics (LNI) - Proceedings
Series of the German Informatics society (GI)

Volume P-113

ISBN 978-3-88579-207-9
ISSN 1617-5468

Volume Editors
Dr. Sören Auer
Universität Leipzig, Institut für Informatik
Johannisgasse 26, D-04103 Leipzig, Germany
e-mail: auer@informatik.uni-leipzig.de
Dr. Christian Bizer
Freie Universität Berlin, Institut für Produktion, Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin, Germany
e-mail: chris@bizer.de
Claudia Müller
Universität Potsdam, Institut für Wirtschaftsinformatik
August-Bebel-Str. 89, D-14482 Potsdam, Germany
e-mail: cmueller@wi.uni-potsdam.de
Dr. Anna V. Zhdanova
ftw. Forschungszentrum Telekommunikation Wien Betriebs GmbH
Donau-City-Straße 1 / 3, A-1220 Wien, Austria
e-mail: zhdanova@ftw.at

printed by Köllen Druck+Verlag GmbH, Bonn

Preface

We are pleased to welcome you to the 1st Conference on Social Semantic Web (CSSW) and wish you a wonderful stay in Leipzig and an enjoyable and rewarding conference participation!

The concept of Social Software was characterizes a variety of software and services on the Web, which enable new ways of communication and social interaction for creating large content bases from a multitude of user contributions. The Semantic Web is an extension of the current Web aiming at enhanced search and navigation facilities and at information integration from multiple sources. ”How the different approaches of Social Software and Semantic Web can be combined in a synergetic way?” - this question appears more and more often in current research and development.

The aim of the Conference on Social Semantic Web is to provide a podium for exploration of Social Software concepts for the bootstrapping of the Semantic Web and lifting Social Software to the semantic collaboration level. CSSW aims at combining three different perspectives on the Social Semantic Web: **the business and entrepreneurial perspective** focusing on the added value of specific social semantic web applications, **the technical perspective** enabling and supporting the exploitation of the “swarm intelligence” of social networks and **the social perspective**, which explores motivations, benefits and emergent effects of the Social Semantic Web. CSSW targets to bring these three perspectives together, to widen existing horizons, to create novel ideas and to find new ways of understanding this emerging field.

The event runs as a two day sub-conference of the Conference on Software, Agents, and Services for Business, Research and E-Sciences (SABRE) and comprises presentations of 9 full and 6 poster and demonstration peer-reviewed papers. In addition, CSSW also includes a keynote by Kingsley Idehen on “Hello Data Web – Exposing the Data Web”, a panel discussion, further late-breaking demo and poster presentations and a special session with 4 papers from the collaborative research project SoftWiki - Distributed, End-user Centered Requirements Engineering for Evolutionary Software Development.

Submissions did not yet reach the level of multi-disciplinarity which we hoped to achieve. The CSSW contributions can be categorized into papers describing (i) models and concepts for social semantic collaboration, (ii) technical support systems, and (iii) applications. Thus, the social sciences and economy communities could have gained more representation. However, the general interest and author feedback encouraged us at reviving CSSW in the next year and we look forward to attracting more multi-disciplinary contributions and fostering the exchange between the “stakeholder” communities of the Social Semantic Web.

We thank the SABRE organizers who helped us tremendously by caring about most of the logistics and overall technical organization. We are grateful to the CSSW keynote speaker and to the members of the program committee who completed the reviews in a quick turnaround time. We also acknowledge the support of GI e.V. and Leipziger Informatik Verbund.

Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova

Programme Committee

Andreas Blumauer, punkt.netServices, Austria
John G. Breslin, DERI, Ireland
Jorge Cardoso, University of Madeira, Spain
Richard Cyganiak, FU Berlin, Germany
Jörg Diederich, L3S, Germany
Sebastian Dietzold, Universität Leipzig, Germany
Orri Erling, OpenLink SW, UK
Kai Fischbach, Universität Köln, Germany
Walter Goix, Telecom Italia, Italy
Andreas Harth, National University of Ireland, Ireland
Tom Heath, Open University, UK
Florian Heidecke, Universität St. Gallen, Switzerland
Ceriel Jacobs, Vakantieland, The Netherlands
Dongwon Jeong, Kunsan National University, Korea
Jason J. Jung, Inha University, Korea
Berit Jungmann, T-Systems MMS, Germany
Markus Krötzsch, AIFB – University of Karlsruhe, Germany
Jens Lehmann, Universität Leipzig, Germany
Peter Mika, Yahoo! Research Barcelona, Spain
Volkmar Pipek, Universität Siegen, Germany
Axel Polleres, Universidad Rey Juan Carlos, Spain
Thomas Riechert, Universität Leipzig, Germany
Harald Sack, Hasso Plattner Institut, Germany
Leo Sauermann, DFKI, Germany
Sebastian Schaffert, salzburgresearch, Austria
Jan Schmidt, Bamberg, Germany
Frank Schönefeld, T-Systems MMS, Germany
Christian Stegbauer, Goethe-Universität, Frankfurt, Germany
Martin Strohbach, NEC Europe, Europe
Kim Viljanen, Helsinki University of Technology, Finland
Jakob Voss, Wikimedia e.V., Germany
Jürgen Ziegler, Universität Duisburg-Essen, Germany

CSSW is supported by the following organizations and projects:

CSSW is a sub-conference of SABRE

GI-Fachgruppe Methoden und Werkzeuge
zur Entwicklung interaktiver Systeme (INSYDE)

Leipziger Informatik Verbund

softWIKI

Project, funded by BmBF as part of the research initiative
“Software Engineering 2006”

Table of Contents

Demonstration and Poster Papers

Hello Data Web - Exposing the Data Web

Kingsley Idehen
OpenLink Software
10 Burlington Mall Road Suite 265
Burlington, MA 01803 USA
kidehen@openlinksw.com

The Web of Documents (Web1.0), and the more recent APIs driven Web of Services (Web 2.0) have collectively become the catalysts of a global data generation, integration, and annotation effort that has paved the way a new dimension of Web interaction commonly referred to as the "Semantic Data Web" (data-web). Thus, it is now possible to interact with the Web in true database fashion.
This talk will provide a general walk-through and live demonstrations that cover of defining elements of the "Semantic Data Web" as they apply to products and emerging market dynamics. It will also provide insights into the applicability of data-web technology, principles, and concepts to realms such as: social-networking, Weblogs, Wikis, Music, Wikipedia, and others Web data sources.

About Kingsley Idehen: Kingsley Idehen is the Founder, President and Chief Executive Officer of OpenLink Software, a leading provider of high-performance Universal Data Access, Data Integration, Hybrid Database Engine technology. In addition to the day-to-day operation and management of OpenLink, he is also responsible for OpenLink's product strategy, vision, and product architecture. Prior to founding OpenLink Software in 1992, he worked for Unisys (in the UK) as a technical specialist focusing on database management and 4GL systems. Kingsley is an industry acclaimed innovator who has been actively involved in database and data access middleware realms since the late 80's. He has been an ardent supporter and backer of Open Data initiatives for many years.

Alternative Searching Services:

Seven Theses on the Importance of "Social Bookmarking"

Dr. Gernot Graefe

Business Development, Corporate
Computing and Communication Lab
Fuerstenallee 11
33102 Paderborn
gernot.graefe@c-lab.de

Dr. Christian Maaß
Dr. Andreas Heß

Lycos Europe
Carl-Bertelsmann-Str. 29
33311 Guetersloh {christian.maass,
andreas.hess}@lycos-europe.com

Abstract: In recent years social bookmark systems like del.icio.us or Furl have become increasingly popular. These systems sometimes are regarded as alternatives to algorithmic search engines like Google. In this paper we develop seven theses on the potential of these systems in order to establish a conceptual basis for future research in this area. Thereby it becomes clear that social bookmarking systems complement rather than threaten algorithmic search engines.

1 Introduction

Together with the exponential growth of the Internet, algorithm-based search engines such as Google or Microsoft Live Search have become the most frequented Web applications. According to conservative estimates 70 to 85% of all information inquiries are serviced by such search engines [HD04]. Their success is mainly based upon their ability to index information automatically and provide it to a great number of users independent of time and place. However, there is empirical evidence that the quality of their search results is rather low. Frequently, only 20 to 45% of the search engine results are relevant results considering the supplied information inquiry [MW03]. One explanation why only such a small portion of the algorithm-based search engine's results are relevant hits are search engine manipulations. As an example one may mention BMW, which after an all too obvious attempt to manipulate Google's search engine algorithms had been temporarily banned from Google's index in early 2006 [BBC06].

Against this background it does not come as a surprise that more and more people wonder whether alternative searching services can compete with algorithm-based search engines with regards to quality of search results [Ne05]. In the media bookmarking systems are already considered as alternatives to Google and other algorithm-based search engines [MNBD06].

However, it is surprising that there are only very few studies on search engine quality [LH07]. This paper shall therefore try to assess the potential of social bookmarking systems. To this end it has to be determined whether social bookmarking systems are in fact alternatives to algorithm-based search engines, and which weaknesses and strengths they possess compared to algorithm-based search engines. First this paper will outline the way social bookmarking systems work and then go on to develop methods and criteria to evaluate the quality of search engines' information retrieval. Finally, seven theses on the importance of social bookmarking systems will be elaborated in order to establish a conceptual basis for future research in this area. The paper ends with a short conclusion.

2 Technological Foundation and Methodical Background

2.1 Characterization of Algorithm-Based Search Engines and Social Bookmarking

In order to access the quality of different searching services we shall first outline the way they work as well as their differences in information retrieval and analysis. Algorithm-based search engines make use of technological resources. So-called Web crawlers automatically analyze the World Wide Web by autonomously following all hyperlinks that are placed on a particular Web page. This allows them to analyze a great part of the Internet and index it for later search inquiries in a rather short period of time. The hyperlinks and page information that the robots can gather are then saved in a special database, the so-called index. This index and the enormous amount of stored data are then used to generate search results for every search inquiry.

Compared to these search engines, social bookmarking systems – such as del.icio.us or Furl – are quite new, and are only discussed in the general public recently. Hence it not surprising that no commonly accepted definition for this term has been established. In principle one may point out that social bookmarking systems are a special form of social software solutions that are used to create social networks and distribute information within these networks [ANRD06].

These social networks play an important role in the context of social bookmark systems as the index is not build by a Web crawler but through the collaboration of the network's members. For this purpose members only need to publish their personal hyperlink collections, or a fraction of it, in the respective bookmark system. Furthermore each hyperlink should be "tagged" with metadata that serves as a description of the particular Link and corresponding Web site [SLRC06]. For example a hyperlink to the "White House" in Washington could be published with the tags "President", "USA", "White House", "sightseeing in Washington" and "George Bush". Through these tags, a search inquiry for the President of the United States could illustrate a connection between the President and the White House, even if the respective Web-document does not reveal such a connection. The technology behind this is based on an analysis of the relation between the tags that reveals the frequency of their joint usage.

The emerging networks of relationships between tags and hyperlinks are called folksonomies. They enable the user to navigate through a collaboratively elaborated index. Figure 1 shows the principles of social networking systems on the basis of the above text. Evidently, these systems are quite different from traditional bookmarking systems.

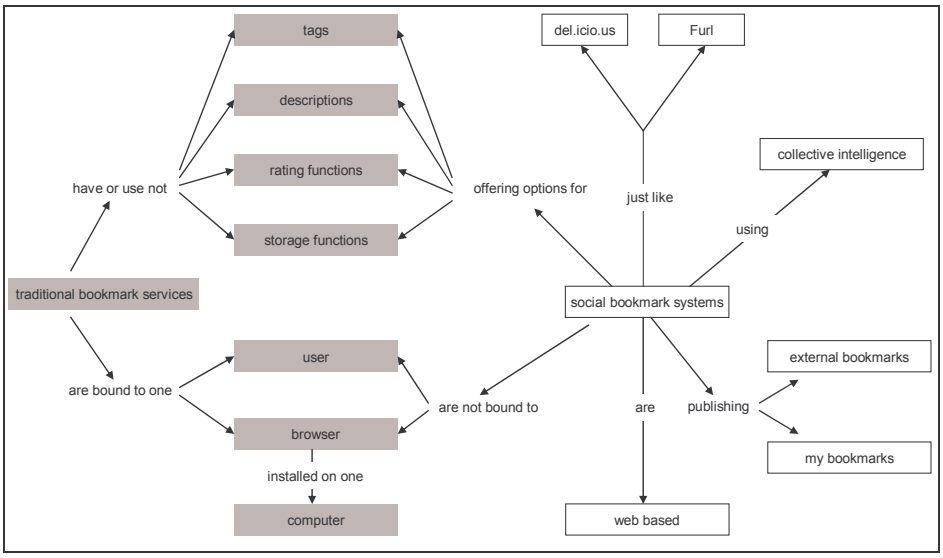


Figure 1: Comparison between traditional and social bookmarking systems [translated from Dö07]

2.2 Current Research on Web-Information Retrieval

Information retrieval constitutes a special area of computer science which deals with the computer-based and content-oriented extraction of information. In order to pay special attention to the peculiarities of retrieving information from the Web and due to the increasing significance of this work, Web-information retrieval now forms an own research area. The assessment of search engine quality is an important part of this research that is usually measured through so called retrieval tests. In these tests search inquiries are sent to selected search engines. Afterwards a panel of experts evaluates the supplied search results according to their relevance [Gr04; Ve06] whereby the so called Precision has become an accepted and commonly used measuring unit. This index number points out the percentage of relevant hits within the total number of search results. However, the mere focus on this percentage is criticized to an increasing degree. It is argued that other measures have to be taken into account in order to gain valuable information on the quality of search engine results [LH07]. Of particular significance are the size and the up-to-datedness of the search index that determines which information the user has access to.

One can conclude from the above text that it is primarily the size and actuality of the search index as well as the search results' relevance that matter in the assessment of search engine quality. Thus these criteria will be used in order to compare algorithm-based search engines and social bookmarking systems in the remainder of this paper. Thereby seven hypotheses will be developed that point out the strength and weaknesses of social bookmarking systems and set a conceptual framework for further empirical studies.

3 The Quality of Social and Algorithm-Based Searching Services

3.1 Size of the Index and the Importance of It Being Up-to-Date

The size of the searching index and its up-to-datedness have already been identified as important indicators of searching service quality. Considering that the indexed part of the World Wide Web currently includes over 11.5 billion Web sites [GS05], it seems arguable that the manual compilation of Web site information done by a community in a social bookmark system could somehow be better than that of automated algorithm-based search engines. This assumption can be strengthened through a direct comparison of the two respective services' search index sizes. Empirical studies show that the indices of algorithm-based search engines of Google, Yahoo and MSN covering nearly 85% of the part of the Internet that can be indexed [Su06]. Hence, several billion Web sites are indexed. In contrast, the leading social bookmarking system in Germany, Mister-Wong, had only indexed about 1.4 million Web sites in early 2007 [Mi07]. Moreover, towards the topics covered, social bookmark systems seem to have a technical and media-oriented focus right now that limits their coverage. Other topics of interest are hardly covered. The existence of such a limited focus can be deduced from an analysis of the most frequently used tags and the respective number of bookmarks in these areas Figure 2 shows an analysis of the bookmark system provided by Lycos-Europe.

Thesis 1: Social bookmarking systems currently cover only a limited number of subjects/ topics on the Internet. However, as social bookmarking systems grow they will continuously widen their range of subjects.

It is not enough, however, to draw one's conclusion on the quality of searching services merely on the basis of their index size as search engine robots index all available Web sites. In contrast social bookmarking systems disregard poor Web sites in a pre-selection. Consequently, the indices of social bookmarking systems will always be smaller simply due to the way they work. Just because of this filter function social bookmarking systems may provide result lists that have a higher Precision with respect to the initial inquiry.

Thesis 2: The smaller index of social bookmarking systems does not correlate with the perceived quality of the respective searching services.

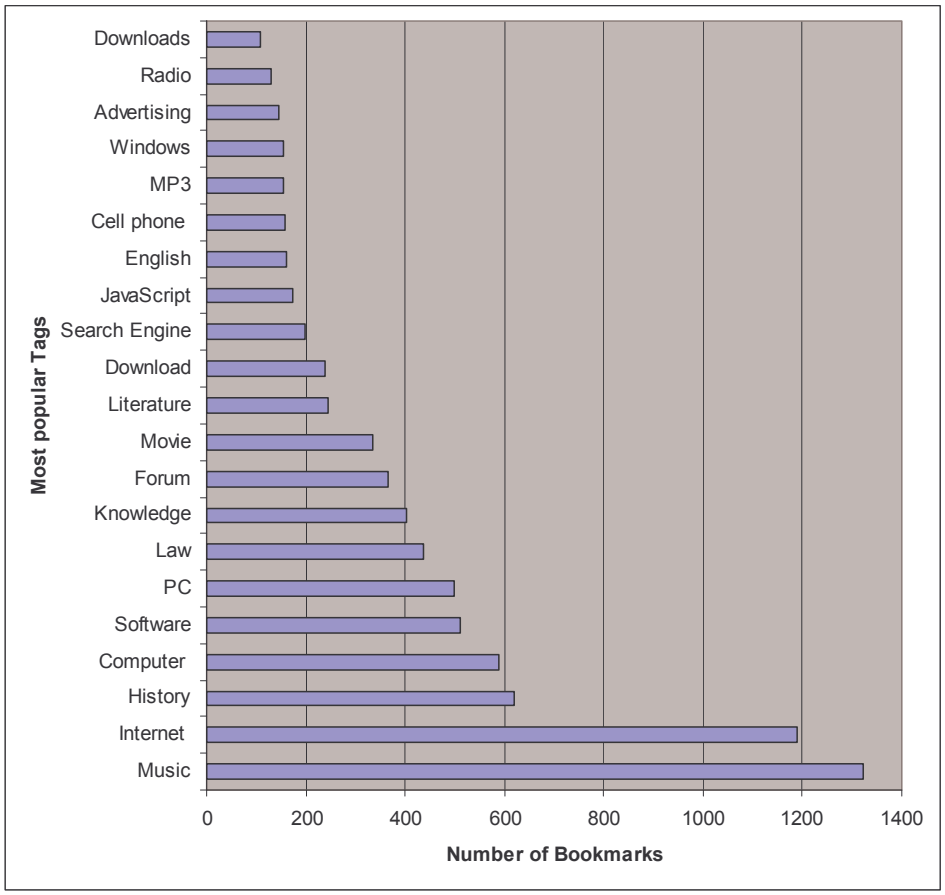


Figure 2: Most frequently used tags in the social bookmarking system Lycos IQ [Ly07]

Aside from the index size, the up-to-datedness of the index is commonly identified as a yardstick for searching service quality. That is because new information is the most looked for information on the Internet, just like business news, sports articles or job opportunities. The value of this information strongly correlates with its up-to-datedness. Thus it seems as a logical approach to consider the frequency in which searching services are updated next.

The average frequency with which search engine providers update their indices is about once every 3.1 days for Google, once every 3.5 days for MSN, and Yahoo updates its index every 9.8 days [LWM06]. Some smaller search engines only update their index in intervals greater than 30 days, which seems unacceptable considering the speed with which content changes on the Internet: 320 million new Web sites are being created per week and within a year about 80% of all Web sites change their link-structure [NCO04].

However, to some extend the high average intervals in which search engine providers update their indices can be attributed to the fact that the providers maintain separate data storages for specific areas of interest [NCO04]. For example, the news index of search engines such as Google and MSN is updated daily. In contrast, the index for image searches is updated in much greater intervals. Though, algorithm-based search engines have a planned renewal strategy for their indices. It is arguable whether social bookmarking systems can deliver recent news to the extent that search engines do. For the integration of current information it is necessary that users add links to their list of bookmarks. As for the short lifespan of a specific piece of current information one may have doubts if users will actually bookmark a specific notice referring to this specific information. It seems much more likely that users will add Web sites to their bookmarks that regularly provide current news. For example the Web site of a news agency will be bookmarked rather than an individual notice on this Web site. In this case it would be impossible to find a dedicated notice by the means of social bookmarking systems.

Thesis 3: Algorithm-based search engines are able to include information into their indices faster and more detailed than social bookmarking systems.

Thesis 4: The greater and more active a community is, the more likely it is to find sites that contain a dedicated notice through search results of social bookmarking systems.

3.2 Relevance of Search Results

Algorithm-based search engines access the relevance of Web sites primarily based upon two factors. Firstly, the analysis of the different elements inside the HTML-code plays an important role in determining the sites' relevance [GC06]. Search engine robots weight the respective elements differently, meaning that not every element has the same impact on the Web site's search engine rating. Take for example keywords that are used in headlines. Based upon the assumption that these keywords summarize the site's content precisely, they are well suited for an assessment of the Web page's contextual relevance. Consequently, text that is declared as a headline in the HTML-code weights heavier in the site's relevance assessment than conventional text does. However, the page content is also very important for the determination of the site's relevance. For this reason many companies try to place frequently used search terms on their Web page. If the search engine locates the respective search term very often, the corresponding Web page will be ranked high in the search engine result list for this particular keyword. Yet the ranking based upon this criterion alone has been subject to several manipulation attempts as popular keywords were systematically integrated into the HTML-code of the Web pages in order for them to receive better ratings.

Secondly, nearly all search engines analyze the link-structure in order to evaluate the contextual relevance and quality of Web pages. It is believed that sites with popular or high quality content receive a higher number of hyperlinks as compared to Web sites with inferior content. In combination with contextual criteria – such as keywords – the link-structure is able to significantly improve the quality of search results.

A few years ago, Google was able to conquer the search engine market due to their implementation of this groundbreaking idea [BP98]. Nevertheless, also this criterion is not resistant to manipulations. Cloaking is one attempted to manipulate the ratings. Cloaking means that special software solutions on Web servers try to distinguish human users from search engine robots. The latter are than forwarded to a special search engine optimized Web page with hyperlinks and keywords that tricks the robot into assuming that it has found a highly relevant page. Without wanting to start a detailed discussion on the problem areas of algorithm-based search engines, we may conclude from the above discussion that algorithm-based search engines depend on criteria that are vulnerable to manipulation attempts. In part, this explains the low Precision of their search results.

Unlike search engines, social bookmarking systems are less vulnerable to manipulation attempts. Here, the contextual relevance of Web pages is not accessed through robots but humans. For the users, it is not the HTML-code elements or the link-structure that affects them to add a Web page to their personal bookmark list. Rather it is the information quality of the respective Web page. For this reason social bookmarking systems base their ratings upon the cumulative number of users that have bookmarked a certain Web page. Therefore Web sites may be ranked high even if there are very few links that lead to this document. In principle, these bookmarking systems are comparable with customer reviews on shopping portals like Amazon that are already used for a fairly long time. This is an important note as these customer reviews are regarded as particularly trustworthy [Eg01; Ni02]. Social bookmarking systems use this factor in order to increase the trustworthiness of their search results as the community disregards low quality content.

Thesis 5: Compared to algorithm-based search engines, social bookmarking systems are far less prone to manipulations. This results in a greater Precision of search inquires.

Thesis 6: Users perceive the search results of social bookmarking systems as more trustworthy than those of algorithm-based search engines.

Still, this rather favorable assessment of social bookmarking systems is put into perspective by the fact that problems in terms of “tagging” are quite frequent. For example, in the course of an analysis of the leading social bookmarking system delicio.us, Lee points out that about 20% of its users do not annotate/ tag any of their bookmarks [Le06]. Moreover, different spellings and subjective combinations of tags lead to more or less diffuse folksonomies. Therefore frequently errors occur while searching for connected issues and subjects. However, it may be assumed that problems originating due to different spellings may be solved by technical means in the near future. For example, algorithm-based search engines are comparing search inquiries with a predetermined vocabulary, immediately identify spelling mistakes and instantly suggest an orthographically correct word to the user. Furthermore several research projects try to address the above mentioned problems by connecting semantic technologies with social software solutions. It is one aim of these projects to automatically extract a Web sites metadata to facilitate the tagging for the user [WZY06].

Thesis 7: The quality of tags will be improved in the near future through semantic technologies.

3 Conclusion

Following the preceding discussion, the strength of social bookmarking systems can be seen in their ability to evaluate the quality of Web sites better than algorithm-based search engines. In addition to that, context relevant connections can be created through metadata/ tags that annotate links and Web pages. While algorithm-based search engines are unable to determine the amplitude and correctness of information encountered on a Web site, the users of social bookmark communities can. They may evaluate a Web site and then share this evaluation with other members of the community. For future retrieval test this means that the currently used technical measures have to be extended by other measures that are able to deliver a better evaluation of the information quality and focusing on the content itself. Nevertheless there are still doubts whether or not social bookmarking systems can compete with algorithm-based search engines with regard to the indexation of current information. It is questionable if users will bookmark Web sites that contain information with a short lifespan. To overcome this problem the automatic generation of metadata could be a visible approach. As shown by Hess et al. such an approach could be a first step to improve and enhance the manual approach of meta data generation in terms of quantity and quality [HMD07].

Upon this background one may resume that social bookmarking systems do not replace algorithm-based search engines. Rather they can be treated as qualitative complement of traditional searching services. Therefore it seems to be a feasible approach for algorithm based search engine providers to integrate the results of social bookmarking systems into their search process, in order to improve the quality of their search results. This approach is already used by search engines like Lycos. However, until now there is no empirical data if such an approach could improve the quality of search results. Furthermore it would be an interesting approach to use the folksonomies generated by the community as a starting point towards the realization of structured vocabulary just like in the field of ontology engineering. As the classic top-down ontology-based approach has not been widely adopted due to its complexity in real-world use, the public has clearly indicated a strong preference for bottom-up approaches using loose folksonomies instead.

4 Acknowledgements

The research presented in this paper was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

Bibliography

[ANRD06] Aleman-Meza, B.; Nagarajan, M.; Ramakrishnan, C.; Ding, L.; Kolari, P.; Sheth, A.; Arpinar, B.; Joshi, A.; Finin, T: Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. In: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, May 23 - 26, 2006. ACM Press, New York, pp. 407-416.

[BBC06] BBC News: BMW given Google 'death penalty', 2006; published on the Internet: http://news.bbc.co.uk/2/hi/technology/4685750.stm (accessed April 9th, 2007).

[BP98] Brin, S.; Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine, Stanford University, 1998.

[Dö07] Döbeli, B.: Social Bookmarking, 2007; published on the Internet: http://beat.doebe.li/bibliothek/w01899.html (accessed April 9th, 2007).

[Eg01] Egger, F. N.: Affective Design of E-Commerce User Interfaces: How to Maximise Perceived Trustworthiness. In: Helander, M. G.; Khalid, H. M.; Tham, M. P. (Eds.): Proceedings of the International Conference on Affective Human Factors Design. Asean Academic Press, London, 2001.

[GC06] Grappone, J.; Couzin, G.: Search Engine Optimization, Sybex, 2006.

[Gr04] Griesbaum. J.: Evaluation of three German search engines: Altavista.de, Google.de and Lycos.de. In: Information Research 9, 4, 2004; published on the Internet: http://informationr.net/ir/9-4/paper189.html (accessed April 9th, 2007).

[GS05] Gulli, A.; Signorini, A.: The Indexable Web is more than 11.5 Billion Pages. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, Chiba, 2005.

[HD04] Hirsh, S.; Dinkelacker, J.: Seeking information in order to produce information: An empirical study at Hewlett Packard Labs. In: Journal of the American Society for Information Science and Technology 55, 9, 2004; pp. 807-817.

[HMD07]: Heß, Andreas; Maaß, Christian; Dierick, Francis (2007): On Semi-Automated Semantic Tagging of Very Short Texts, Lycos Research Paper 1/2007, Gütersloh 2007.

[Le06] Lee, K.: What Goes Around Comes Around: An analysis of del.icio.us as social space. In: Proceedings of the 20th anniversary conference on Computer supported cooperative work, 2006; published on the Internet: http://delivery.acm.org/10.1145/1190000/1180905/p191-lee.pdf?key1=1180905

&key2=7444766711&coll=&dl=ACM&CFID=15151515&CFTOKEN=6184618 (accessed April 9th, 2007).

[LH07] Lewandowski, D.; Höchstötter, N.: Web Searching: A Quality Measurement Perspective. In: Spink, A.; Zimmer, M. (eds.): Web Searching: Interdisciplinary Perspectives. Springer, Dordrecht 2007.

[LWM06] Lewandowski, D.; Wahlig, H.; Meyer-Bautor, G.: The Freshness of Web search engine databases. In: Journal of Information Science 32, 2, 2006; pp. 131-148.

[Ly07] Lycos: Link library, 2007; published on the Internet: http://iq.lycos.de/lili/srch/ (accessed April 9th, 2007).

[Mi07] Mister-Wong: Mister-Wong – Startseite, 2007; published on the Internet: http://www.mister-wong.de/ (accessed April 9th, 2007).

[MNBD06] Marlow, C.; Naaman, N.; Boyd, D.; Davis, M.: HT06, Tagging Paper, Taxonomy, Flickr, Academic Article. To Read. In: Proceedings of the seventeenth conference on hypertext and hypermedia, 2006; pp. 31-40.

[MW03] Machill, M.; Welp, Carsten: Wegweiser im Netz, Bertelsmann Stiftung, Gütersloh, 2003.

[NCO04] Ntoulas, A.; Cho, J.; Olston, C.: What's new on the web? The evolution of the web from a search engine perspective. In: Proceedings of the 13th WWW Conference, New York, 2004; published on the Internet: www2004.org/proceedings/docs/1p1.pdf (accessed April 9th, 2007).

[Ne05] Neymanns, H.: Suchmaschinen: Das Tor zum Netz, Bundestagsfraktion der Grünen, Berlin, 2005; published on the Internet: http://www.gruene-bundestag.de/cms/publikationen/dokbin/63/63265.pdf (accessed April 9th, 2007).

[Ni02] Nikander, P.: Trustworthiness as an Asset. In: Schubert, S.; Reusch, B.; Jesse, N. (Hrsg.): Informatik bewegt, Tagungsband der 32. Jahrestagung der Gesellschaft für Informatik e.V., 2002; pp. 100-105.

[SLRC06] Sen, S.; Lam, S.; Rashid, A.; Cosley, D.; Frankowski, D.; Osterhouse, J.; Harper, F.; Riedl, J.: Tagging, communities, vocabulary, evolution. In: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, Alberta, 2006; pp. 181-190.

[Su06] Sullivan, D.: Nielsen NetRatings Search Engine Ratings, Search Engine Watch, 2006; published on the Internet: http://searchenginewatch.com/showPage.html?page=2156451 (accessed April 9th, 2007).

[Ve06] Véronis, J.: A comparative study of six search engines, 2006; published on the Internet: http://www.up.univ-mrs.fr/veronis/pdf/2006-comparative-study.pdf (accessed April 9th, 2007).

[WZY06] Wu, X.; Zhang, L.; Yu, Y.: Exploring social annotations for the semantic web. In: Proceedings of the 15th International Conference on World Wide Web, New York, 2006; pp. 417-426.

Collaborative Web-Publishing with a Semantic Wiki

Rico Landefeld, Harald Sack
Friedrich-Schiller-Universität Jena,
D-07743 Jena
rico.landefeld@takwa.de, sack@minet.uni-jena.de

Abstract: Semantic wikis have been introduced for collaborative authoring of ontologies as well as for annotating wiki content with semantic meta data. In this paper, we introduce a different approach for a semantic wiki based on an ontology meta model customized especially for the deployment within a wiki. For optimal usability client-side technologies have been combined with a simple semantic query language. Text fragments of a wiki page can be annotated in an interactive and rather intuitive way to minimize the additional effort that is necessary for adding semantic annotation. Thus, the productivity and efficiency of a semantic wiki system will open up for non expert users as well.

1 Introduction

The very first browser of the World Wide Web (WWW) provided a function that soon sank into oblivion again: web pages could not only be read, but also written and thus be changed directly. Some years ago, wiki systems [LC01] picked up that idea again by providing the possibility for each visitor to change the content of wiki pages. Wiki systems are lean content management systems that administrate HTML documents. The user of a wiki system is able to generate or change wiki documents only by using the facilities of a simple web browser. In this way, wiki documents are developed and maintained collaboratively by the community of all users. Wiki systems don't give formal guidelines for generating or structuring their content. This lack of formal rules might have been responsible for their fast growth of popularity as can be seen, e.g., in the free online-encyclopedia Wikipedia[1]. On the other hand, if a wiki system is growing as rapidly as Wikipedia does, lack of formal rules necessitates frequent restructuring to keep the content well arranged and usable.

Typical wiki systems only provide a limited number of functions for structuring the content. As a rule users create special pages with overviews or class systems for structuring the wiki content. But the maintenance of this manually created categorization system becomes rather expensive. Moreover, it stimulates misusage, as e.g., you may find many categories in Wikipedia that have been created to subsume entities that share merely one special feature [VKV+06]. Similar problems have been reported for intranet wikis [BG06]. In general, most of the mentioned problems in wikis can be reduced to the fact that their content is encoded in HTML (Hypertext Markup Language) or some simplified version of it.

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

HTML only formalizes formatting and (limited) structuring of text without the possibility of formalizing any semantics that is required for automated aggregation and reuse of data.

Semantic Wikis try to combine wiki systems with semantic technology as a building block of the currently emerging semantic web [BLHL01]. They connect textual content with a knowledge model by formalizing the information of a wiki page with a formal knowledge representation language. In this way, the content of wiki pages becomes machine readable and even machine understandable. Semantic wikis show one possible way to overcome the aforementioned problems related to traditional wikis in general while at the same time enabling collaborative generation and maintenance of formal knowledge representations (ontologies). But, the arbitrary wiki user is not an expert knowledge engineer. Therefore, usability and "ease of us" become a rather important factor for designing the user interface of a semantic wiki.

Current projects have chosen different ways to deploy formal knowledge representations within a wiki. From our point of view the ratio of cost and effect is most important. The cost refers to the cognitive and factual work that the user has to invest to generate and maintain semantic annotations. On the other side, the effect subsumes all the advantages that the user might get from a system that deploys this semantic annotation. Cognitive and factual work is mostly determined by the design of the user interface and the underlying ontology meta model of the wiki. At the same time the semantic expressiveness of the annotations determines the efficiency of the achieved functionality. Therefore, the ontology meta model of a semantic wiki represents a compromise between complexity and expressiveness. In addition, the integration of semantic annotations into wiki systems demands new concepts of user interaction that help to limit the necessary effort. Existing semantic wiki systems have several deficiencies: either, their underlying ontology meta model is mapping elements of the knowledge representation language directly to wiki pages, or they are using a simplified ontology meta model that results in rather limited semantic functionality. Most projects are based on traditional wiki systems and therefore inherit also their user interaction facilities.

We propose a semantic wiki concept that combines the following three concepts: a simplified ontology meta model especially customized to be used within a wiki system, a WYSIWYG-Editor (What you see is what you get) as a user interface for both text- and ontology editing, and preferably a most simple semantic query language. A prototype of our semantic wiki *Maariwa*[2] has been successfully implemented. The paper is organized as follows: Section 2 covers related work and in particular discusses ontology meta models and user interaction concepts. In Section 3 we introduce the semantic wiki project Maariwa, while Section 4 resumes our results and discusses future work.

2 Related work

Besides the extension of traditional wikis with semantic annotation, we also have to consider approaches for collaborative authoring of ontologies based on wiki technology that

[2] http://ipc755.inf-nf.uni-jena.de:8081/Maariwa

date back to a time before the semantic web initiative even started (cf. [FFR97, SRKK97, ACFLGP01]). We therefore distinguish two different semantic wiki approaches depending on their focus either on textual (wiki) content or (formal) knowledge representation. The *Wikitology* paradigm [DRR+05b] refers to wiki systems acting as a user interface for collaborative authoring of ontologies. There, a wiki page represents a concept and hyperlinks between wiki pages represent relationships between concepts. Thus, the wiki system acts merely as tool to manipulate the ontology. In difference, so called *ontology-based wiki systems* are semantic wiki systems that focus on textual content, while using knowledge representations to augment navigation, searchability, and reusability of information.

Another differentiating factor is determined by the adaption of the ontology meta model for the use within the wiki and the coverage of the underlying knowledge representation languages (KRL). The ontology meta model of a semantic wiki defines a mapping between the elements of the KRL and the application model. Moreover, it determines the semantic expressiveness of annotation and serves as a basis for querying information. Semantic annotation can be maintained together with or separate from the textual wiki content. Next, we will introduce and discuss relevant semantic wiki implementations and their underlying ontology meta model.

Platypus Wiki [CCT04] is one of the earliest semantic wiki implementations. It maps a wiki system to an RDF (resource description framework) graph. Wiki pages represent RDF resources and hyperlinks represent RDF properties. Semantic annotation is maintained together with the textual wiki content within a separate text field as RDF(S) (RDF schema) [BG04] or OWL (web ontology language) [MvH03] in XML serialization format.

Rhizome [Sou04] supports semantic annotations by using a special Wiki Markup Language (WikiML). The entire wiki content including text, structure, and meta data internally is encoded in RDF. In contrast to traditional wiki systems, Rhizome supports a fine-grained security model. Besides manipulation of meta data Rhizome does not offer any functionality that utilizes this semantic annotation.

Rise [DRR+05a] is customized for requirement analysis in software engineering. It is based on an ontology that represents different document types and their relationships. Templates determine structure and relationships of wiki pages that represent instances of a document type. The Rise ontology can be extended by adding new templates. Semantic annotations can be edited with an extended WikiML and are used for consistency check and navigation.

Semantic MediaWiki (SMW) [VKV+06, KVV05] is an extension of the well known MediaWiki [3]. The online-encyclopedia Wikipedia is the most prominent example of a MediaWiki application. SMW aims to improve structuring and searchability of the Wikipedia content by deploying semantic technologies. Therefore, SMW tries to follow Wikipedia's user interface to attract a broad user community. SMW extends the WikiML with attributes, types, and relationships. To represent classes, SMW utilizes existing Wikipedia categories. By assigning a wiki page to a given category it becomes an instance of the class being represented by this category. In addition, SMW provides a set of units of measurement and customizable data types. Attributes, types, and relationships are represented

[3] http://www.mediawiki.org/

by own wiki pages. Semantic search is implemented in SMW with a proprietary query language (WikiQL) that closely reflects the annotation syntax.

Makna Wiki [DPST06] uses RDF triples (subject, predicate, object) for annotating wiki pages. RDF triples can be added to a text page by using a customized WikiML or within a separate form. RDF triples' subjects or objects refer to textual wiki pages that represent a concept each. Makna Wiki only supports maintenance and manipulation of instances, but no class definitions. It extends JSPWiki[4] and its WikiML with typed links and literals. Makna Wiki's semantic annotation is utilised for navigation based on RDF triples and for limited semantic search (e.g., for searching instances of classes with distinct properties).

IkeWiki [SGW05] tries to bring together application experts and knowledge engineers. Therefore, the user interface offers seperate views for textual content and semantic annotation. The annotation editor supports the assignment of classes to wiki pages and typed links for representing relationships. Furthermore, classes, properties, and resources can be freely created and manipulated. The ontology meta model closely reflects the underlying KRL (RDF(S) and OWL). The user interface for editing meta data supports automatic completion of terms.

SweetWiki [BG06] combines social tagging [GH06] and semantic technologies into what they call *semantic tagging*. Wiki pages are annotated with user tags that form not only a collective index (a so called *folksonomy* [Van05]), but a formal ontology. This is achieved by regarding each user tag as a concept of an ontology. Relationships between concepts are not determined by users, but by designated experts. The user does not interact with the ontology directly, but is merely able to create and to assign user tags. For the user, there is no distinction between instances and classes, because tags can represent both. Sweet Wiki's user interface provides a WYSIWYG editor for manipulating the wiki content.

Besides the above mentioned projects there exist several alternative semantic wiki implementations that can also be arranged within the framework given by our examples ranging from wikitologies to ontology-based wikis. As a rule, early implementations such as Platypus strictly separate textual content from knowledge representations, while later projects (besides IkeWiki, which separates annotations from content) integrate knowledge representations and textual content by utilizing customized WikiML. Above all, IkeWiki and MaknaWiki provide dynamic authoring support. The ontology meta model of PlatypusWiki, MaknaWikia and IkeWiki merely provide a direct mapping of the underlying KRL without any covering. Only MaknaWiki offers (limited) semantic search facilities. If elements of RDF(S) or OWL are utilized directly, RDF query languages such as SparQL [PS07] can be applied. IkeWiki enables data export via SparQL without providing a search interface. MaknaWiki and Platypus use elements of RDF(S) and OWL without making any use their semantic expressiveness.

Contrariwise, SMW deploys a simplified ontology meta model based on OWL-DL with an easy to use query language (compared to SparQL) The SMW user interface keeps the connection between classes and their attributes covered, but offers no further editing assistance to the user (besides the provision of templates). Sweet Wiki's ontology meta model does not distinguish between classes, instances, datatype properties, or object properties –

[4] http://jspwiki.org/

everything is mapped on tags. Therefore, information can only be provided via tag search or via direct SparQL queries with the consequence that information being distributed over several wiki pages can not be queried.

We propose the semantic wiki *Maariwa* that utilizes an ontology meta model covering the underlying OWL-Lite language that enables information reuse as well as semantic queries over distributed information. In the following chapter we introduce Maariwa's underlying concepts and give a brief sketch on its implementation.

3 The Maariwa concept – architecture and implementation

Maariwa is a semantic wiki project of the FSU Jena with the objective of implementing an augmented wiki system that enables simultaneous creation and manipulation of textual content and ontologies. Maariwa's semantic annotation is utilized to put augmented navigation and semantic search into practice for reuse and aggeregation of the wiki content. In Maariwa ontologies are used to structure the textual wiki content for providing access paths to the knowledge being represented in the wiki. Maariwa's access paths can be addressed via *MarQL*, a simple semantic query language, to enable semantic search. We therefore refer to Maariwa's underlying concept as *ontology-based web publishing*. Maariwa is geared towards user communities without expert knowledge in knowledge representation. Furthermore, Maariwa facilitates access by utilizing a WYSIWYG-editor for text and ontology manipulation.

3.1 Maariwa's Ontology Meta Modell

The Ontology meta model of Maariwa is designed to enable simple and efficient searchability, as.e.g. answering questions like "What physicists were born in the 19th century" or"Which cities in Germany have more than 100.000 inhabitants?". To answer these questions, the ontology meta model has to provide the semantic means to express these queries, while on the other hand it must be simple enough that the arbitrary user without expert knowledge can comprehend it. Tab. 1 shows a subset of OWL-Lite elements that are utilized for Maariwa's ontology meta model. We refer to datatype properties and object properties as attributes and relationships. Furthermore, both attributes and relationships are not defined as global entities but only local within their classes. This enables attributes and relationships with the same name in different classes without worrying the non expert user with naming conflicts. Attributes are determined by a datatype with an optional measuring unit. Only numbers, strings, and dates are considered for datatypes.

By integrating the ontology meta model into the wiki system, wiki pages can be annotated with concepts of ontologies to formalise their content. Classes, individuals, and sets of individuals can be described by wiki pages. A page that describes a class is associated with attributes, relationships, and superclasses. Pages that describe individuals are associated with one class at least. To denote a set of individuals, a page has to be associated with a

OWL-Lite element	Maariwa element
class	class
datatype property	attribute
object property	relationship
class instance	wiki page representing a class being istantiated as an instance-page
subClassOf	superclass /subclass
individual	wiki page describing an individual, being an instance of one or more classes
datatype property instance	attribute value of an instance-page
object property instance	relationship value of an instance-page

Table 1: Mapping of OWL-Lite elements to Maariwa's ontology meta model.

MarQL expression (see section 3.4). Typed links between pages may refer to relationships. Classes may use instance-pages as simple tags by associating neither a class hierarchy nor relationships or attributes.

3.2 Integration of semantic annotation and textual content

The Maariwa editor is designed to minimize the user's additional effort for adding and maintaining semantic annotations. Repeated input of text or ontology concepts is avoided most times simply by copying existing concepts from the page context. New concepts are created in dialogue with the user. Thus, Maariwa enables the development of textual and ontological resources in parallel. Maariwa provides two alternatives for creating semantic annotations: concepts of an ontology can be created from textual content of a wiki page, while on the other hand concepts might be defined first providing a textual description in the wiki page afterwards. Schema and instance data can be manipulated in parallel without interrupting the editing process of the wiki page (see Fig. 1). Maariwa's WYSIWIG-editor is implemented as so called *Rich Internet Application* [Loo06] that provides desktop functionality for web applications and adopts the role of the traditional WikiML-based user interface. In Maariwa, semantic annotations are directly displayed within the wiki page (see. Fig. 2). Different colors denote the semantics of typed hyperlinks. Text that contains attribute values as well as links that represent relationships is highlighted. In addition, tooltips (i.e., pop-up information windows) display the semantic of an annotated text fragment if is is touched with the mouse pointer. Navigation within class hierarchies and class relations is enabled with a superimposed class browser. The semantic annotation of each wiki page can be separately accessed and exported in RDF/XML encoding via an own URL. Also export and import of ontologies as a whole is supported.

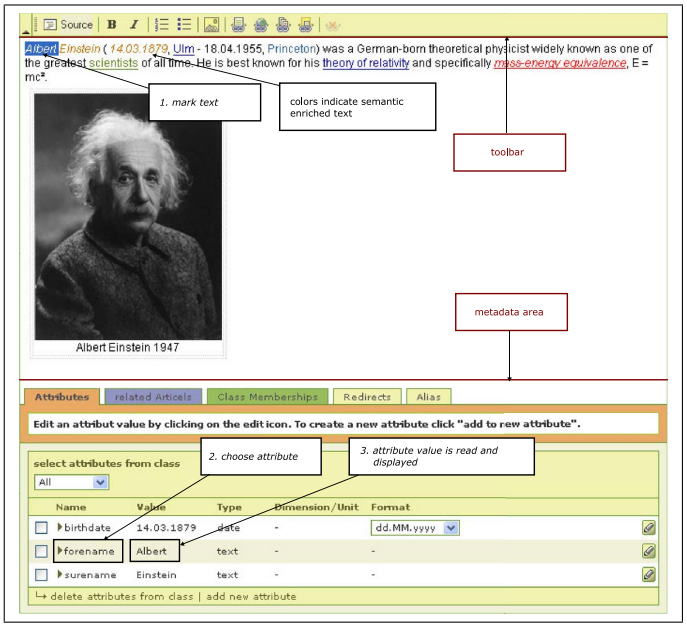


Figure 1: Editing of a Maariwa wiki page with textual content (above) and ontologies (below).

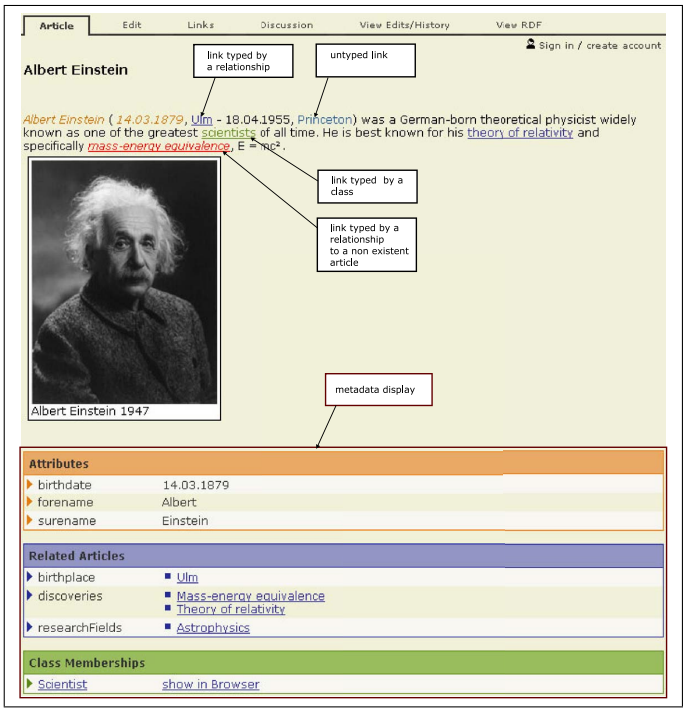


Figure 2: A wiki page with semantic annotation in Maariwa.

3.3 The MarQL semantic query language

The syntax of SparQL reflects the characteristics of the RDF data model. RDF data are represented as triples and the RDF document can be interpreted as a graph. SparQL traverses the RDF graph and as result delivers the nodes that satisfy the constraint given in the SparQL query. Because RDFS and QWL are based on the RDF syntax SparQL can also be used to query RDFS- and OWL-files, but without exploiting their semantic expressiveness.

MarQL is a customized semantic query language for the Maariwa ontology meta model. MarQL syntax does not refer to RDF triples but directly addresses ontology elements such as classes, attributes and relationships. The underlying RDF encoding of the data remains hidden. In comparison to SparQL, the syntax of MarQL is much more compact but less flexible. MarQL only implements a fixed set of query patterns. A MarQL query results in a set of wiki pages that refer to individuals, which satisfy the constraints of the MarQL query. The structure of a MarQL query can be shown with an example: the expression *Scientist.institution.location.country = Germany* refers to wiki pages about scientists that work at an institute being located in Germany. *Scientist* refers to a class with a relationship *institution*. Relationships can be applied recursively and are denoted as a path expression. In this way, *institution* and *location* are connected. This means, that there must exist a class, which is target class of a relationship with *institution*, while in addition having a relationship with *location* . In the same way *location* and *country* are connected, while *country* can either be an attribute or a relationship and therefore *Germany* might denote an individual or an attribute value. MarQL provides logical operators as well as string operators and operators for comparison. E.g., the query *City.population* $\geq$ *100.000* results in a set of all wiki pages that describe cities with more than 100.000 inhabitants, or *Scientist.birthdate* $<$ *1.1.1900 AND Scientist.birthdate* $\geq$ *1.1.1800* results in a set of wiki pages with scientists that are born in the 19th century. The latter example results in a list of instances of a class and can also be referred to as a simple *tag*.

3.4 Maariwa architecture and implementation

Maariwa does not extend any traditional wiki system but is a proprietary development based on Java. The application core of Maariwa implements a service level that realizes a wiki system as a set of loosely coupled services (see Fig. 3). Besides data management and revision control of the wiki pages and the related ontologies, keyword search and semantic search are also implemented as services. MarQL queries are translated into SparQL queries. Manipulation of the ontologies in the editor is organized in different dialog levels. Within a dialog updates of one or more objects can be performed. These updates can either be revoked or they will also be adopted in the subjacent levels. Therefore, the service level offers cascading manipulation levels that implement this functionality with the help of local copies and snapshots. Backend data processing is achieved with a relational database management system. The service level stores ontology elements in various RDF triple stores and all other objects in separate database tables. The service level is used by the components of the web front-end that constitute Maariwa's user interface. Web server

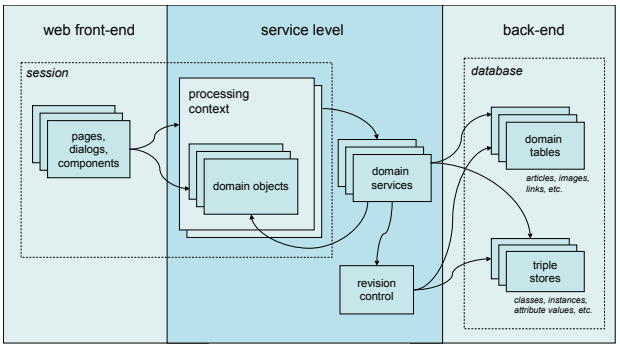


Figure 3: Architecture of the Maariwa system.

and browser client communicate asynchronously via AJAX [Gar05] to achieve better usability. The WYSIWYG-editor represents the text of the wiki page as XHTML fragment and is based on Javascript.

As a rule, wiki systems deploy a revision control system (RCS) to prevent abuse of unprotected write access. Maariwa adapts this RCS for ontologies, too. The RCS maintains two levels: schema level and instance level. The schema level comprises all changes on classes, relationships and attributes, while the instance level covers changes of individuals including their attribute values and relationship values. Both levels are tightly coupled, because each schema update might have effects on all derived individuals. Therefore, a schema version additionally includes all updates on the instance level that occurred since the last update of the schema. Each version of a wiki page besides the update of the page content also contains updates of the associated concepts.

4 Conclusion

In this paper, we have introduced the semantic wiki approach Maariwa based on an ontology meta model customized especially for the deployment within a wiki. For optimized usability recent client-side technologies have been combined with a simple semantic query language. The user can annotate text fragments of a wiki page in an interactive and rather intuitive way to minimize the additional effort that is necessary for adding semantic annotation. Thus, the productivity and efficiency of a semantic wiki system will open up for non expert users as well. The ontology meta model enables the formulation of access paths to wiki pages as well as the reuse of already implemented relationships in the underlying knowledge representation. The simple query language MarQL uses Maariwa's annotations and the contained access paths for implementing a semantic search facility. Ontology meta model and query language are synchronized to support annotations on different levels of expressiveness. Thus, enabling simple tags as well as complex ontologies with attributes and relationships.

Currently, Maariwa is extended to include meta data also directly within the textual wiki content, as e.g., tables with self-adjusting data aggregations. Also natural language processing technology is considered to be utilized in the WYSIWYG-editor for automated suggestions as well as for translating natural language queries into MarQL. For deploying a semantic wiki, the user considers always the ratio of cost and effect that is caused by the additional effort of providing semantic annotation. In doing so, it is not important whether the creation of semantically annotated textual content or the creation of mere ontologies is focussed, but how both can be integrated within an system that provides a reasonable and efficient interface for user access.

References

[ACFLGP01] J. C. Arpírez, O. Corcho, M. Fernández-López, and A. Gómez-Pérez. WebODE: a Scalable Workbench for Ontological Engineering. In *1st Int. Conf. on Knowledge Capture (KCAP01).Victoria. Canada*, 2001.

[BG04] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3c recommendation, RDF Core Working Group, W3C, 2004.

[BG06] M. Buffa and F. Gandon. SweetWiki: semantic web enabled technologies in Wiki. In *Proc. of the 2006 international symposium on Wikis*, pages 69–78, 2006.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[CCT04] S. E. Campanini, P. Castagna, and R. Tazzoli. Platypus Wiki: a Semantic Wiki Wiki Web. In *Semantic Web Applications and Perspectives, Proc. of 1st Italian Semantic Web Workshop*, 2004.

[DPST06] K. Dello, E. Paslaru, B. Simperl, and R. Tolksdorf. Creating and using Semantic Web information with Makna. In *Proc. of the 1st Workshop on Semantic Wikis – From Wiki To Semantics*. ESWC2006, 2006.

[DRR+05a] B. Decker, E. Ras, J. Rech, B. Klein, and C. Höcht. Self-organized Reuse of Software Engineering Knowledge Supported by Semantic Wikis. In *Workshop on Semantic Web Enabled Software Engineering (SWESE), at ISWC 2005, Galway, Ireland*, 2005.

[DRR+05b] B. Decker, J. Rech, E. Ras, B. Klein, and C. Hoecht. Self-organized Reuse of Software Engineering Knowledge supported by Semantic Wikis. In *Workshop on Semantic Web Enabled Software Engineering (SWESE), at ISWC 2005*, 2005.

[FFR97] A. Farquhar, R. Fikes, and J. Rice. The Ontolingua server: A tool for collaborative ontology construction. *Int. Journal of Human-Computer Studies*, 46(6):707–727, 1997.

[Gar05] J. J. Garrett. Ajax: A New Approach to Web Applications, 2005. `http://www.adaptivepath.com/publications/essays/archives/000385.php`.

[GH06] S. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Sciences*, 32(2):198–208, April 2006.

[KVV05] M. Krötzsch, D. Vrandecic, and M. Völkel. Wikipedia and the semantic web - The missing Links. In *Proc. of the 1st Int. Wikimedia Conf., Wikimania*, 2005.

[LC01] B. Leuf and W. Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.

[Loo06] C. Loosley. Rich Internet Applications: Design, Measurement, and Management Challenges. Technical report, Keynote Systems, 2006.

[MvH03] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language: Overview. Working draft, World Wide Web Consortium, March 2003.

[PS07] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF (Working Draft). Technical report, W3C, March 2007.

[SGW05] S. Schaffert, A. Gruber, and R. Westenthaler. A Semantic Wiki for Collaborative Knowledge Formation. In *Semantics 2005, Vienna, Austria*, 2005.

[Sou04] A. Souzis. Rhizome Position Paper. In *Proc. of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

[SRKK97] B. Swartout, P. Ramesh, and T. Russ K. Knight. Toward Distributed Use of Large-Scale Ontologies. In *Symp. on Ontological Engineering of AAAI. (Stanford, California, March)*, 1997.

[Van05] T. Vander Wal. Folksonomy Explanations, 2005. `http://www.vanderwal.net/random/entrysel.php?blog=1622`.

[VKV+06] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. of WWW 2006, Edinburgh, Scotland*, 2006.

Weaving Space into the Web of Trust: An Asymmetric Spatial Trust Model for Social Networks

Mohamed Bishr

Institute for Geoinformatics,
University of Muenster,
Robert-Koch-Str. 26-28,
48149 Muenster, Germany
m.bishr@uni-muenster.de

Abstract. The proliferation of Geo-Information (GI) production in web-based collaboration environments such as mapping mashups built on top of mapping APIs such as GoogleMaps API poses new challenges to GI Science. In this environment, millions of users are not only consumers of GI but they are also producers. A major challenge is how to manage this huge flow of information and identify high value contributions while discarding others. The social nature of the collaborative approaches to GI provides the inspiration for innovative solutions. In this paper, we propose a novel spatial trust model for social networks. This model is part of our research to formalize the spatio-temporal regularities of trust in social networks. The presented model provides a metric for trust as a proxy for GI quality to assess the value of collaborative contributions. We also introduce the underlying network for the model, which is a hybrid network structure for collaborative GI applications based on affiliation networks and one-mood continuous trust networks. Trust calculation in the affiliation network takes into account the geographic distance between the actors and their information contributions –also known as events–, while the one-mood network does not. This leads to an asymmetric model with respect to the representation of space.

Keywords: Trust, space, spatial, quality, proxy, social networks, trust model.

1 Introduction

The notion of trust in computer science has many definitions depending on the application domain. It is a descriptor of security and encryption [KA98], a name for authentication methods or digital signatures [An01], a factor in game theory [MRS03], and a motivation for online interaction and recommender systems [AH98]. In this paper we are concerned with the web of trust and particularly consider the social aspect of trust in web based social networks (WBSN) and define it as a representative of personal values, beliefs and preferences. This type of social trust has been studied in previous work such as [Go05, MSC04, RAD03, ZL04]. As further explained in this paper, the majority of this research has been on the usage of trust as a measure of the quality and relevance of information in social networks of online communities.

In [BK07] we discussed two different examples of collaborative GI environments, Openstreetmap.org and Wayfaring.com. Each has its own unique aspects as a collaborative GI application. Wayfaring.com in particular demonstrates mapping

mashups built over mapping data exposed through new Mapping APIs such as Google and Yahoo Mapping APIs. The wealth of information layers added to these mapping data are characterized by locality, as users contribute local knowledge and by breadth of scope with a variety of information such as pictures, restaurant reviews, jogging tracks and much more. Such layers of knowledge are otherwise hard to acquire on such a large scale and they make the underlying datasets ever richer and more valuable. We identified some shortcomings associated with these collaborative GI environments due to the transformation of users from GI content consumers to GI content producers. This transformation certainly raises new challenges with respect to the management of GI and its semantics. Also in [BK07] we presented a scenario where we pointed out the increasing demands for up-to-date geographic information in the context of road navigation for truck drivers. To overcome the above challenges, it is important to understand which users provide valuable contributions and which do not, both in data and its metadata. A major obstacle to this is the lack of quality measures of collaboratively generated GI since traditional quality measures (lineage, accuracy, consistency and completeness) are almost impossible to determine in such application environments.

In this paper, we present a novel approach to use trust in social networks as a proxy of GI quality. The hypothesis of this approach is that trust as a proxy for geospatial information quality has a spatio-temporal dimension that needs to be made explicit for the development of effective trust based GI quality metrics. The spatial dimension acts as a measure of the confidence in the trust rated events. Our goal is a spatio-temporal model of trust in social networks. This spatio-temporal trust model should identify trusted information contributions by users as well as identify and reaffirm trust in those users who contributed and continue to contribute information. In our approach, space and time are orthogonal dimensions and therefore the proposed model separates between them. In this paper, we only focus on the spatial dimension of trust in social networks.

The asymmetric spatial model of trust in social networks introduced in this paper is based on a hybrid social network model consisting of two interlinked networks:

- a two-mood non-dyadic affiliation network represented by a bipartite graph. In the affiliation network, we refer to the information contributions of the actors as events. The links between actors and events in the affiliation network are weighted by the distance between the location of the actors (presumably home base address) and the location of the events.
- A one-mode network of actors as one-mood continuous trust-rating network [Go05].

The proposed model is said to be asymmetric because space is represented only in the affiliation network in terms of geographic distance between actors and events, while the confidence in one-mood network trust ratings remains insensitive to space.

2 Trust In Social Networks

Trust from a sociological point of view is a prerequisite for the existence of a community, as functioning societies rely heavily on trust among their members [Co01, Fu96, Us02]. “The existence of trust is an essential component of all enduring social relationships” [Se97 p. 13]. Online communities much like real life communities rely on trust that is either implicit between the community members or explicit where users rate each others with a numerical measure of trust as studied in [Go05, ZPM05].

The inclusion of a computable notion of trust into social networks requires a simple definition of the term that preserves relevant properties of trust in our social life [Go05]. A simple yet inclusive definition of trust, which we adopt for our work is *"Trust is a bet about the future contingent actions of others"* [Sz99]. There are two components to this definition, belief and commitment [Go05]. The person (trustor) believes that another person (trustee) will act in a certain way. Then, the trustor commits to an action based on that belief. Four properties of trust on WBSN are also identified [Go05], namely:

- Transitivity
- Composability
- Personalization
- Asymmetry

These basic properties of trust are well grounded in our adopted definition of trust and will be considered in the formal model discussed in this paper. For further details on these trust properties and their relation to the adopted definition we refer the reader to [Go05, RAD03, ZL04]. An additional property of trust that we endorse for our work is presented in [Sz99], trust is essentially about people, and behind complex constructs, that we may vest trust in, always lays individuals whom ultimately we endow with trust. In Sztompka's [Sz99] view to trust Lufthansa for a flight to Tokyo means you trust the pilots, cabin crew, and ground crew and so on. It easily follows that if you trust certain information, you trust the person or persons behind this information. This allows for a possible transition from trust between individuals to trust the information provided by those trusted individuals.

Trust in WBSNs is a measure of how information produced by some network users is relatively valuable to others [Go05, ZPM05, ZL04]. Trusted users tend to provide more useful and relevant information compared to less trusted or un-trusted users. With the lack of traditional information-quality attributes (lineage, accuracy, consistency and completeness) in collaborative environments, we propose to use trust as a proxy for geospatial information quality. Quality is a subjective measure here (and always, to some extent); if some trust-rated geospatial information is useful and relevant to a larger group of users, it can then be assumed to have a satisfactory quality in a more objective sense.

3 Spatial Aspects of Trust in Social Networks

The evolution of social networks with respect to the spatial dynamics, directly relating network evolution to constraints defined by the geographic space was studied in [MM05]. In their study of trust [BK95, BK96] provide evidence about the effects of social network structures on trust. Networks of high density compel actors to confirm each other's opinions rather than argue about differences. Therefore, actors in dense networks tend to have more extreme opinions about the trustworthiness of others compared to actors in less dense networks. These opinions can veer towards trust or distrust. The dynamics of the emergence of these opinions are not explained. However, the network dynamics approaches discussed in [NBW06,Wa04,WS98] provide interesting insights on the role network dynamics can play in enhancing our understanding the trusting behavior of actors in social networks.

"Homophily is the principle that a contact between similar people occurs at a higher rate than among dissimilar people. The pervasive fact of homophily means that cultural, behavioral, genetic, or material information that flows through networks will tend to be localized" [MSC01 p.416]. In their study of homophily, [MSC01] asserts that space is a

very basic source of homophily. People are more likely to have contact with those others who are geographically closer than with others further away. A justification for this natural tendency is provided in [Zi49] as a matter of effort: it takes more effort (time, attention, etc.) to connect to those who are faraway than those who are closer. Many studies have illustrated this correlation between geography and establishment of connections between people [Ca90,Ga68]. Furthermore, factors that seem trivial such as arrangement of streets do have a direct effect on the formation of relatively weak ties and the potential for strong friendship formation [HW00, Su88].

With the proliferation of communications, the web and its pervasiveness one can expect that these technologies have had an effect on the extent to which geography affects social networks. These new technologies have apparently loosened the effects of geography by lowering the effort involved in creating and maintaining contacts [KC93]. Despite this loosening effect of new technologies [We96a] shows that social networks maintained through technological means still show geographic patterns. Also [Ve83] shows that geographic proximity is still the single best determinant of how often friends socialize together. In [We96b] it also becomes clear that new communication technologies have allowed people greater affordances in making homophiles relations through other dimensions (e.g., those with pet hobbies can form relations over the internet with others anywhere in the world more often than in the past). "these technologies seem to have introduced something of a curvilinear relationship between physical space and network association, with very close proximity no longer being so privileged over intermediate distances but both being considerably more likely than distant relations"[MSC01 p.430]. Also [MSC01] continues to assert that geography seems more important in determining the *"thickness"* of a relationship, which pertains to its multiplexity and frequency of contact.

Although the effects of the spatial dimension on trust particularly in social networks are not fully understood, this *"thickness"* of the relationships described by McPherson [MSC01] is a strong determinant of trust [Se97, Sz99]. Other factors such as the outcome of each encounter between actors (positive/negative experiences), the nature of those encounters and their importance to the parties involved also strongly affect trust [Bu02, Se97, Sz99]. One notices the multidimensional nature of social relations [Wa04], where space is a factor that helps shape the current nature and the future of those relations and consequently the levels of trust inherent in those relations.

Temporal and spatial aspects of trust in organizations are discussed in [Rä04] by grounding his discussions in the Greek notions of chronos(clock time)/kiros(right moments) for time and their spatial counter parts chora(abstract space)/topos(concrete place). He tries to lay the philosophical foundations of how those notions affect trust particularly in time management and virtual organization settings. However, no conclusions are made about a direct effect of geographical space on trust. In the context of studying networks of organizations [NE92] argues that partners in close proximity to each other are almost often preferred partners, which means that network embeddedness (the over all network structures and mutual relations of actors) is ultimately affected by geographical space. In addition, [Bu02] suggests a direct link between geographical space and trust in social networks. He uses geographic distance as an indicator of network density among firms. His assumption is that there is a higher probability that firms located geographically closer together will have ties to common third parties and also that these third parties have more contacts among each others making the social

network dense in social ties as a result of geographic space which eventually fosters trust in the social network.

4 An Asymmetric Spatial Trust Model for Social Networks

In our view trust is inferred about people not about inanimate objects [Sz99]. We are adopting here a social definition of trust as we previously discussed. To trust information contributed by the actors is to implicitly trust actors who contributed this information. In that sense there is a certain unity or tie between actors and information they contribute, our proposition is that geographic distance affects the confidence we have in a user's trust rating of a certain information entity. We suggest that this tie can be viewed as a physical affiliation in graph theoretic terms. Initially we propose to view the actors contributing information as a bipartite graph (Fig. 1). We will refer from now on to information contributions where a user is reporting a roadblock for example as "events". In a bipartite graph, there are two subsets of nodes and all lines are between nodes belonging to the subsets. Formally, in a bipartite graph, nodes in a given subset are adjacent to nodes from the other subset, but no node is adjacent to any node in its own subset [WF94]. Our hybrid network consists of both an affiliation network and a one-mood [Wf94] actors network. Before discussing the model, we address two basic assumptions that are derived from our previously discussed study of the prior work on the spatial aspects of social networks and trust:

- **assumption 1:** trust in events increases as the number of users tagging the event increases.
- **assumption 2:** we have higher confidence in trust ratings of actors who are geospatially closer to events. In that sense distance in our model affects the *confidence* in a trust rating of an information entity.

Model Description

In social network terms, the resulting bipartite graph of actors and contributions is a two mood non-dyadic affiliation network. Actors are possibly affiliated to many contributions and contributions are possibly contributed by many actors. Hence, our affiliation network consists of a set of actors and a set of events. We observe two sets of nodes representing the two moods of the network (Fig. 1):

***N*:** is the set of all actors who are contributing information (events) to the network $N = \{n_1, n_2, ..., n_g\}$. Those actors can be viewed in terms of the subset of events to which he contributed.

***M*:** Is the set of all events that are contributed by actors *n*, $M = \{m_1, m_2, ...m_h\}$ similar to the members of the set *N*, events can also be viewed in terms of the subset of actors who contributed to these events.

The affiliation network shown in (Fig. 1) is said to be non-dyadic. In non-dyadic networks the affiliations formed around the events in this network consist of subsets of actors such that event $M_1 = \{n_2, n_5, n_6\}$. Therefore, relations are not anymore between pairs of nodes as opposed to one-mood networks. Of course in such a network actors can also be viewed as subsets of events such that actor $N_2 = \{m_1, m_3, m_5\}$. However, at this stage of the model we are interested only in the events as subsets M_h. From this initial

setting, our first measure of the quality of the events can be conveyed with the graph theoretic nodal degree measure.

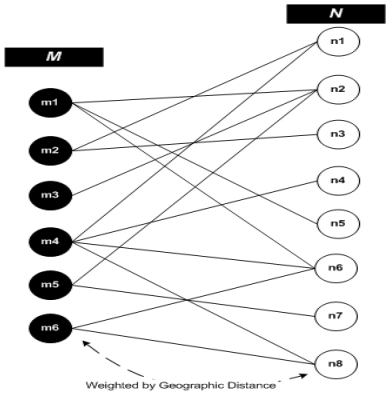


Fig. 1. The affiliation network of actors and the events (information they contributed). The links representing the affiliation is weighted by geographic distance to incorporate distance effects in the model

For a graph with *g* nodes, the degree of a node denoted $d(m_h)$ is simply the number of lines that are incident with it. The nodal degree can range from 0 if no lines are incident with a given node to $g-1$ if a node is adjacent to all other nodes in a graph. In our bipartite graph of actors and events one can measure the nodal degrees of the events, which essentially would represent the number of members in the subset M_h. One should note that in the proposed model the nodal degree will never be 0 and will have a minimum of 1 since each event only exists in response to an initial user contribution. The nodal degree here is a simple albeit a very telling measure of the importance of a particular node/event [WF94]. In our case it tells us how many actors/users have actually bookmarked a certain location as a roadblock for example. Considering such a measure, the higher the nodal degree of the event the more we can reaffirm trust in the information provided since this means that more people are reaffirming it. Given the property of trust we adopted earlier from [Sz99] this is a highly acceptable measure.

However, when calculating the nodal degree in the standard graph theoretic method it is assumed that all the links from actors to events are of equal importance. This is contrary to reality since some users will be inclined to provide more relevant information. As previously discussed, the hypothesis underlying this model is that the spatial dimension is a strong governing factor to this inclination to trust users in providing more accurate information. This view is supported by evidence from our discussion of the spatial aspects of trust in social networks. Hence, we propose a representation of space, particularly distance as a weight to the links when calculating the nodal degrees. Events whose users are closer to the event location (e.g. higher relative proximity) receive a better rating of trust than these events whose users are further away. This spatial nodal degree measure will be denoted $\grave{d}(m_h)$. We assume the location of the actor to dynamically his location when reporting an event (i.e. collected from a mobile device, personal reporting) and the distance from the event location to be

the direct geographic distance in a straight line. In the nodal degree calculation every link between an actor and an event is counted as one. For example the nodal degree of the event m_1 is 3 and m_4 is 4 (Fig. 1). For our distance adjusted nodal degree for event nodes, we take a naïve representation of distance by dividing each nodal degree $e=1$ by the corresponding distance c, thus:

$$\grave{d}(m_h) = \sum_{i=1}^{k} \frac{e}{c_i} \quad (1)$$

In our case e is always 1, thus:

$$\grave{d}(m_h) = \sum_{i=1}^{k} \frac{1}{c_i} \quad (2)$$

The spatial nodal degree makes the trust inference about the events spatially sensitive, hence adding effects of geography to the social space. However, we don't intend to use this measure alone as a measure of trust in information quality provided by the user. Rather the intention of our model is to make the effects of social trust on the model more explicit. Therefore, we proceed to discuss the second element of the model that integrates continuous trust networks into the model [Go05].

As a common analysis method of affiliation networks, the bipartite graph of the affiliation network can be folded into two one-mood networks one which presents the connections between actors based on affiliation with events, and another which represents the connections between events based on affiliation with actors. At this stage we are not trying to define the meaning of both networks. The distinction here is that the one-mood network of actors that will be discussed next is not resulting from the folded bi-partite graphs. Rather it is a continuous trust network of actors where ties represent trust ratings on a scale of 1-10 [Go05]. Our model requires the underlying system to provide facility for these direct user-to-user ratings forming a standard one-mood network of actors (Fig. 2)

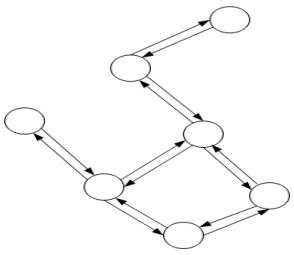


Fig. 2. : The one-mood trust network. This trust network is by definition directed and represents the asymmetry property of trust earlier discussed (rating from A→B ≠ B→A)

The resulting network is presented in (Fig. 3). This form of network is not strictly an affiliation network. In an affiliation network the actors are not connected, but rather

connect through affiliation with events. In our model, actors have two types of connections:

- Connection with events (the affiliation network)
- Connections among themselves (one-mood continuous trust network).

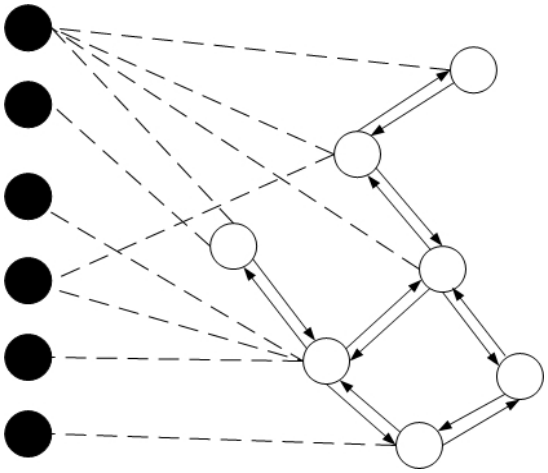


Fig. 3. The depicted network structure represents the fusion of both the one-mood network (white ←→white) and the affiliation network (white→black). The model depends on the interplay between those two networks, as distinct networks within the same structure.

The continuous trust network is used to provide a more explicit account of trust which is bound to a formally well defined notion of trust [Go05, ZL04]. This is best explained by an example. In the network in (Fig.1) Alice (n_2) has contributed to the event m_3 (let us assume it is a roadblock). When calculating the nodal degrees we weight the links by distance. This is assuming that all actors in the set $M_3 = \{n_2, n_5, n_6\}$ contributing to the event m_3 are similar in every respect. This assumption does not hold true when actors are part of a continuous trust network themselves. In that case, trust ratings of the individual actors can be used as weights to reaffirm the ties between the actors and the events through differentiation between actors by their reputation inherited from their trust levels within the social network. Alice in that example is a member of a social network and she has been rated by her peers on the trust scale. We then need to identify what is the over all trust rating of Alice inferred from the social network? This trust rating will then be used as the weight for Alice when making the overall trust rating for an event to which Alice contributed.

The problem of trust inference in the one-mood network is different from [Go05], where the problem of trust is focused on determining how much does A (source) trust G (sink) (Fig. 4(a)) who are not directly connected. In our problem Alice is the sink G (Fig. 4(b)) for which we would like to find a trust rating from the adjacent neighbors and there is no source to infer trust from along the paths. The solution to our inference problem can follow a different methodology albeit with a slightly similar averaging technique as in [Go05].

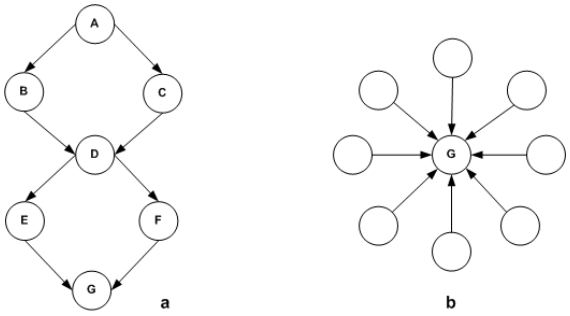


Fig. 4. In [Go05] (a) the problem is computing trust along the paths from source A to sink G. In our case (b), the problem is computing trust to the sink at one degree of separation

Trust rating for a certain sink node (n_g) will be taken as the average of trust ratings directed inwards from the nodes immediately adjacent to it at one degree of separation. That is to say, the geodesic (social) distance $d(i,g)$ is always equal to 1 between the neighbors and the sink (G). The sink node n_g would have a trust rating of t_{n_g} defined by Equation 3 letting N be the number of adjacent nodes. Equation 3 respects the properties of trust discussed earlier, particularly composability, asymmetry and personalization. However, transitivity is not relevant at this stage, since we compute trust at one degree of separation from the sink.

$$t_{n_g} = \frac{\sum_{n_i \in adj(n_g), i=1}^{\mathrm{N}} t_{n_i n_g}}{\mathrm{N}} \quad (3)$$

The integration of the spatial nodal degree as a trust measure (Equation 2) and the trust measure from the one-mood actors network (Equation 3) as a weight for the actors will then provide our global trust level in a certain event. For an event m_h the final trust rating t_{m_h} is defined by Equation 4.

$$t_{m_h} = \sum_{i=1, g=1}^{k} \frac{t_{n_g}}{c_i} \quad (4)$$

Equation 4 establishes a trust metric for information contributed by actors. This metric has an explicit account of geographic distance inherited through the affiliation network as well as an explicit account of trust inherited through the one-mood network. An important decision we made is what we refer to as the asymmetry of the model. This asymmetry of the model is revisited in the last section.

5 Conclusions and Future Work

In this paper we proposed an approach to formalize how distance affects the confidence we might have in trust of certain information entity reported by moving social network agents. A major challenge we have is in establishing a distance threshold after which the confidence effect would have a stable maximum value. Generally, defining such a threshold should involve real world analysis of actual data to fine tune the current model. Also the representation of distance as introduced in the model should be normalized. However the distance measure as represented is remains relevant, as suggested by some recent research [MWB07].

An important point is our choice not to make the continuous one-mood trust network spatially sensitive (asymmetry of the model). This is contrary to evidence suggesting that the continuous one-mood trust network is sensitive to the spatial dimension [Bu02, NE92]. It is clear then that our model will have to be symmetric. That is to account for space in the one-mood continuous trust network when doing trust calculations to the sink. However, accounting for space in both networks at this stage will make the results of any analysis very hard to interpret since it will be difficult to disentangle the effects of space in the one-mood network from those of space in the affiliation network. Hence, the best way forward was to study both as flip sides of a coin before integrating the spatial dimension into both networks in a unified model if so proves more accurate.

Currently our model depends on structural social network analysis methods such as the nodal degree and averaging of trust values of adjacent nodes. However, a modern and emerging view in the science of networks is network dynamics [BC03,Wa04]. In this view, analysis for networks including social networks without a corresponding theory of dynamics is essentially un-interpretable [Wa04]. Actors are not static nodes in a network structure that is the an embodiment of their relations. Actors are human beings moving in space and time, doing activates and responding to a changing environment. In studying trust, our model assumes a network of a given structure as a still image and tries to propagate trust across this network. From a network dynamics perspective we are interested in how trust evolves and develops over the network in both space and time and how trust propagation ultimately affects the structure of the networks. In other words, our presented spatial trust model will have to be accompanied by a corresponding theory of dynamics of trust on social networks.

References

[AH98] Abdul-Rahman, A. and Hailes, S.: A distributed trust model. ACM Press New York, NY, USA (1998) 48-60

[An01] Ansper, A., et al.: Efficient long-term validation of digital signatures. Springer (2001) 402–415

[BC03] Barabási, A. L. and Crandall, R. E.: Linked: The New Science of Networks. Vol. 71. AAPT (2003) 409

[BK07] Bishr, M. and Kuhn, W.: Geospatial Information Bottom-Up:A Matter of Trust and Semantics (In print). Accepted for publishing in AGILE. Springer-Verlag, Aalborg, Denmark (2007)

[BK95] Burt, R. S. and Knez, M.: Kinds of Third-Party Effects on Trust. Vol. 7 (1995) 255

[BK96] Burt, R. S. and Knez, M.: Trust and third-party gossip. In: Kramer, R. and Tyler, T. (eds.): Trust in Organizations: Frontiers of Theory and Research. Sage Publications, inc (1996) 68-89

[Bu02] Buskens, V. W.: Social networks and trust. Springer (2002)

[Ca90] Campbell, K.: Networks past: a 1939 Bloomington neighborhood. Soc. Forces **69** (1990) 139-155

[Co01] Cook, K. S.: Trust in society. Russell Sage Foundation New York (2001)

[Fu96] Fukuyama, F.: Trust: The Social Virtues and the Creation of Prosperity. Vol. 457 (1996)

[Ga68] Gans, H.: People and plans: essays on urban problems and solutions. New York (1968)

[Go05] Golbeck, J. A.: Computing and Applying Trust in Web-based Social Networks. Department of Computing, Vol. PhD. University of Maryland (College Park), Maryland (2005)

[HW00] Hampton, K. and Wellman, B.: Examining Community in the digital neighborhood:Early results from Canada's wired suburb. In: Ishida, T. and Isbister, K. (eds.): In Digital Cities: Technologies, Experinces and future Perspectives. Springer Verlag, Heidelberg (2000)

[KC93] Kaufer, D. and Carley, K.: Communication at a Distance:The effect of print on Socio-Cultural Organization and Change. Lawrence Erlbaum, Hillsdale, NJ (1993)

[KA98] Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol. RFC 2401, November 1998 (1998)

[MRS03] McCabe, K. A., Rigdon, M. and Smith, V. L.: Positive reciprocity and intentions in trust games. Vol. 52 (2003) 267-75

[MSC04] McGuinness, D. L., da Silva, P. P. and Chang, C.: IWBase: Provenance Metadata Infrastructure for Explaining and Trusting Answers from the Web. (2004)

[MSC01] McPherson, M., Smith-Lovin, L. and Cook, J. M.: Birds of a Feather:Homophily in Social networks. Annual Review of Sociology **27** (2001) 415-444

[MM05] Metcalf, S. and Paich, M.: Spatial Dynamics of Social Network Evolution. Vol. 51 61801

[NBW06] Newman, M. E. J., Barabási, A. L. and Watts, D. J.: The structure and dynamics of networks. Princeton University Press (2006)

[NE92] Nohria, N. and Eccles, R. G.: Networks and organizations: structure, form, and action. Harvard Business School Press (1992)

[Rä04] Rämö, H.: Moments of trust: temporal and spatial factors of trust in organizations. Managerial Psychology **19** (2004) 760-775

[RAD03] Richardson, M., Agrawal, R. and Domingos, P.: Trust management for the semantic web. Second International Semantic Web Conference, Sanibel Island, Florida (2003) 351–368

[Se97] Seligman, A. B.: The Problem of Trust. Princeton University Press (1997)

[Su88] Sudman, S.: Experiments in measuring neighbor and relative social networks. Social Networks **10** (1988) 93-108

[Sz99] Sztompka, P.: Trust: A Sociological Theory. Cambridge University Press (1999)

[Us02] Uslaner, E. M.: The Moral Foundations of Trust. Cambridge University Press (2002)

[Ve83] Verbrugge, L.: A Research note on adult friendship contact:A dyadic perspective. Soc. Forces **62** (1983) 78-83

[WF94] Wasserman, S. and Faust, K.: Social Network Analysis: methods and applications. Cambridge University Press (1994)

[Wa04] Watts, D. J.: Six Degrees: The Science of a Connected Age. W. W. Norton & Company (2004)

[WS98] Watts, D. J. and Strogatz, S. H.: Collective dynamics of'small-world'networks. Vol. 393 (1998) 409-10

[We96a] Wellman, B.: Are personal communities local? A Dumptarian reconsideration. Soc. Networks **18** (1996a) 347-354

[We96b] Wellman, B., et al.: Computer Networks as Social Networks:Collaborative Work, Telework and virtual community. Annual Review of Sociology **22** (1996b) 213-38

[ZPM05] Zaihrayeu, I., da Silva, P. P. and McGuinness, D. L.: IWTrust: Improving User Trust in Answers from the Web. Springer (2005)

[ZL04] Ziegler, C. N. and Lausen, G.: Spreading Activation Models for Trust Propagation. The IEEE International Conference on e-Technology, e- Commerce, and e-Service, , Taipei, Taiwan (2004)

[Zi49] Zipf, G.: Human behavior and the Principle of Least Effort. Addison Wesley, Menlo Park, CA (1949)

[MWB07] Mok, D., Wellman, B. and Basu, R.: How Much Did Distance Matter before the Internet? Interpersonal Contact and Support in the 1970s. Social Networks (2007)

A Prototype to Explore Content and Context on Social Community Sites

Uldis Bojārs
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Benjamin Heitmann
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Eyal Oren
Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Abstract: The SIOC Ontology can be used to express information from the online community sites in a machine-readable form using RDF. This rich data structure allows us to easily analyse and extract social relations from these community sites. We use SIOC information to analyse the social relations between users through the content that they create. We introduce metrics for social neighbourhood and social reputation, formally expressed as SPARQL queries over SIOC data. Finally, we demonstrate these algorithms in our Social SIOC Explorer prototype.

1 Introduction

Online community sites (such as blogs, wikis and bulletin boards) are playing an important role in keeping people informed and facilitating communication on the Web. Some of these site are more centralised, others are more decentralised, but from an abstract perspective all such communities play a similar role: they allow users to gather together online, create content and enter into discussions about their topics of interest.

Often, discussions range over several of these communication channels. People try to keep up with these discussions by following web feeds, however, current feed formats only allow to see a single stream of content (posts or comments) and does not provide enough information on the social aspect of these online discussions, e.g., who replies to posts by this author and how often or how to find more information about the author of these posts.

SIOC (Semantically Interlinked Online Communities) [BHBD05] is an RDF vocabulary which allows online community sites to export their data in a semantically rich and interlinked manner. Until now, SIOC browsing interfaces have mainly focused on exploring the content itself or on providing a graph view of this information [BBP06, HO07]. The work presented in this paper looks further: towards the social relations manifested by the content created by users across the online community sites.

1.1 Social context in online communities

Feed readers and search engines for online communities in general focus on the content created in these communities. However, the social context of the message and its author are equally important. Therefore, we would expect that adding the social context to the browsing process would significantly enrich the user experience in selecting information.

Some of these relations may be expressed explicitly, through personal FOAF[1] profiles in RDF or through lists of friends on the social networking sites. However, these explicit relations require constant maintenance and tend to become outdated over time.

Therefore we turn towards object-centered sociality [KC97] and evidence of the connections between people gleaned from the content they create, co-annotate, and reply to. These collaborations uncover the implicit relations between people, but are typically ignored by metadata exporters and feed reader applications.

The SIOC Ontology gives us access to exactly this structure of the content created in the community, and can therefore be used for such object-centered social analysis. By combining all three aspects of the online community: the user-created content, the explicitly defined relations and the social relations derived from the content, we can provide a unified user interface that facilitates both the exploration of the content and the social browsing.

1.2 Outline

To realise these goals, we have started at the first chronological step, by enriching the social community sites with SIOC RDF exporters that automatically create high-quality data; having that data relieves us from screen-scraping and reconstructing the relations between online information. We have then proceeded to consolidate and extract the implicit social relations from that data, to finally build an exploration interface that uses both the community content and the social relations in that community:

1. Produce high-quality data SIOC RDF data from these sites
2. Consolidate distributed SIOC data and extract implicit social relations
3. Build a prototype SIOC explorer for exploring social communities

The paper is structured according to these three steps: we provide a short overview of the SIOC standard in Section 2 and describe the data sources and our integration methodology in Section 3. Section 4 describes the extraction of social context from the integrated community information, and Section 5 introduces the Social SIOC explorer prototype to browse and explore community content. We summarise and conclude in Section 7.

[1] `http://foaf-project.org/`

2 Semantically interlinked online communities (SIOC)

The SIOC initiative aims to ease the integration of online social community information [BHBD05]. SIOC provides on the one hand a unified ontology to describe such online communities, and secondly, several exporters from that translate community information from weblogs, forums, and mailing lists into RDF using the SIOC vocabulary.

The SIOC Core Ontology[2] defines the main concepts and properties required to describe information from online communities on the Semantic Web. The main terms in the SIOC core ontology are shown in Figure 1. Users create content Items (e.g., Posts) that reside in Containers (e.g., Posts in Forums) on data Spaces (e.g. Sites).

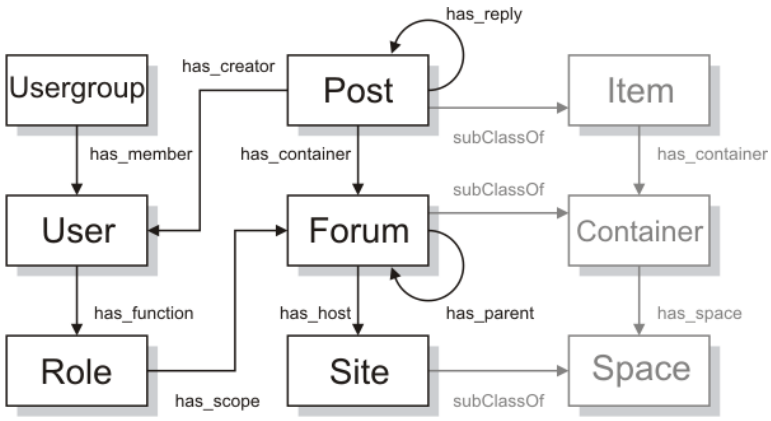


Figure 1: Main SIOC classes and properties

These classes allow to structure the information on online community sites and distinguish between different kinds of objects. Some of the main SIOC properties also play an important role in the context of this paper. `sioc:has_reply` property links replies (e.g., comments) to the original posts while `sioc:has_creator` and `foaf:maker` properties links all the user-created content to more information about its authors. Together these properties form a core set of relations for extracting the social neighbourhood information described in Section 4.

One of the problems with combining social media data is in knowing which accounts users hold on different social media sites. SIOC attempts to solve this by re-using the FOAF (Friend of a Friend) vocabulary which can describe links between a person and accounts it holds in a distributed manner. By combining SIOC with FOAF data we can also re-use the information from personal FOAF profiles, e.g., the `foaf:knows` relationships.

SIOC RDF data export tools[3] have been developed for different types of online community sites and content including *blog engines* such as WordPress, DotClear, b2evolution; *forum*, *bulletin board* and *CMS engines* such as Drupal or PhpBB, *microblogging* tools such as

[2] `http://www.w3.org/Submission/sioc-spec/`
[3] `http://sioc-project.org/exporters`

Twitter and Jaiku, and *mailing lists and IRC conversations*.

These tools make available an RDF representation of all the essential information about the content and users creating it on a Social Media site. This RDF representation contain semantically rich and inter-linked information which can be crawled using a generic RDF crawler by following rdfs:seeAlso links between different data pages. Typically such a crawler would be able to start at site's main SIOC profile page and retrieve RDF about all the publicly available content that a site contains. More advanced tools can be used to incrementally crawl the new information as it appears on a site.

3 Data and methodology

SIOC data were crawled[4] from ten online community sites (namely weblogs) from the list of SIOC-enabled sites[5]. FOAF profiles about the persons, authors of `sioc:Posts`, were retrieved. There was a difficulty retrieving FOAF profiles for people because homepages and blogs often do not have a machine-readable indication of where FOAF profiles can be found. As a result, finding the FOAF profiles was mainly a manual task.

The combined dataset contained $\pm$ 118.000 triples, including 62 `sioc:Users` (users registered on these community sites), 5815 posts and comments (an average of nearly 600 posts per weblog which is reasonable since SIOC exports the full history of a weblog), and in total 3310 `foaf:Persons` (more than the 62 users, since this number also includes all comment authors). The data furthermore contained 1421 unique homepages (because those are usually required for each comment), only 6 unique email addresses (since these are for privacy reasons by default not exported), and 91 hashcodes of email addresses (optionally exported for weblog owners or comment authors that supply email addresses).

The crawled data from disparate online community sites had to be integrated and consolidated. Using RDF and the SIOC vocabulary, the integration part was straightforward: using for example the NTriples RDF serialisation, multiple datasets can simply be concatenated and the duplicate lines can be removed. This was done using the Redland rapper[6] tool for converting the SIOC data from RDF/XML into NTriples, and the Unix tools `cat` and `uniq` for concatenation.

Next, the data needed to be consolidated. We used the inverse functional properties in the dataset to consolidate similar resources with different URIs. However, given the size of the integrated dataset direct OWL reasoning over this dataset was not practically possible. Instead, we separated schema-level reasoning from instance-level reasoning. We first recursively extracted and fetched a list of all used schemas in the dataset (including SIOC, FOAF, WordNet, XML Schema, and many others) and loaded all schemas (totaling $\pm$83.000 triples) into the OWL reasoner, namely the OWLIM extension [KOM05] to Sesame [BKvH02]. We then, after reasoning, extracted a list of all inverse functional properties in these schemas (16 in total), and wrote a manual algorithm to process the

[4] `http://rdfs.org/sioc/applications/#crawling`
[5] `http://esw.w3.org/topic/SIOC/EnabledSites`
[6] `http://librdf.org`

Listing 1: Query for user's direct neighbourhood (L1)

```
SELECT DISTINCT ?relatedPerson
WHERE {
  ?relatedPerson rdf:type foaf:Person .
  ?profileA foaf:knows ?relatedPerson .
  ?profileA foaf:holdsAccount personA . }
```

instance-level triples. The algorithm grouped synonym resources, known to be the same through their inverse functional properties, into a common bucket, and, for each bucket, rewrote all statements to use one canonical URI for all the synonym URIs.

4 Extracting social context

We extract two types of social contextual information from the online community sites. On the one hand, we extract the social neighbourhood of each site member, formed by the set of people that he knows directly or indirectly via online interaction. On the other hand, we extract the social reputation of each member, based on their community involvement, on their activity level and on their connectedness to their peers.

4.1 Extracting social neighbourhood

We define three levels of neighbourhood that can be extracted from the data, either explicitly stated, or implicitly derived from having a small social distance or from co-authoring or co-producing community content. Each neighbourhood is defined as a SPARQL query on our structured SIOC data, leading to a clean and formal definition of these relations and enabling straightforward analysis on actual community data, namely by executing these queries on the instance data. We consider three neighbourhood levels from `personA` to others.

Level 1 consists of the explicitly stated `foaf:knows` relations from `personA` to others. Listing 1 shows this neighbourhood defined as a query, using the direct `foaf:knows` links between people. Since SIOC data uses a `foaf:holdsAccount` property to link a `foaf:Person` and a `sioc:User` account together, the query has to traverse those links. By default, `foaf:knows` relations are not defined to be symmetric; if we want to treat them as such we can union the results with the same query but in the opposite direction.

Level 2 is an implicit neighbourhood relation, meaning that the neighbourhood is not explicitly stated but rather constituted by people who have replied to content created by `personA`. Listing 2 shows this neighbourhood expressed as a query. Note that the query uses a "select distinct"; to rank the neighbourhood by the amount of replies, a count of results returned by an ordinary "select" can be used.

Listing 2: Query for user's indirect neighbourhood (L2)

```
SELECT DISTINCT ?relatedPerson
WHERE {
  ?p rdf:type sioc:Post .
  ?p sioc:has_creator personA .
  ?p sioc:has_reply ?reply .
  ?reply foaf:maker ?relatedPerson . }
```

Listing 3: Query for user's indirect neighbourhood (L3)

```
SELECT DISTINCT ?relatedPerson
WHERE {
  ?p rdf:type sioc:Post .
  ?p sioc:has_reply ?reply1 .
  ?reply1 foaf:maker personA .
  ?p sioc:has_reply ?reply2 .
  ?reply2 foaf:maker ?relatedPerson . }
```

Expanding the neighbourhood further through the usage of shared objects, we define the Level 3 neighbourhood as all people who participated in the same conversations as `personA` (having all replied to the same post), as shown in Listing 3.

The extraction queries presented here should be considered preliminary results. Of course, there are many more types of queries that could be run on the dataset, connecting users through the tags and topics of posts, through a shared geographical region, and more. It should be clear though, that by using SIOC data we are able to very easily extract a particularly defined social relation, without the need for manual data collection, laborous cleansing and integration.

4.2 Extracting social reputation

After extracting the social neighbourhood of a site member, we extract his social reputation. The way in which a user of multiple sites produces and publishes content, and in which his involvement is received by other users in his community are indicators of the social reputation of a site member. Since we cannot measure the reputation directly (social community sites typically do not include "reputation" feedback, as implemented on peer-to-peer trading sites such as eBay.com or Amazon.com), we approximate a user's reputation through his activity level and community involvement.

We associate the activity level (R1) of a user with the number of posts on his own site and with the number of replies that he has written on other sites. A lower number of posts and replies generally indicates a lower activity level. Listing 4 shows the queries used to extract the number of published posts and replies. In the SIOC data, original posts have a `sioc:has_creator` and a `sioc:has_container` property, whereas replies have a `foaf:maker` and are connected to a post through a `sioc:has_reply` property.

Listing 4: Query for user's activity level (R1)

```
SELECT COUNT DISTINCT ( ?publishedPost )
WHERE {
  ?publishedPost rdf:type sioc:Post
  ?publishedPost sioc:has_creator personA .
  ?publishedPost sioc:has_container ?postsContainer . }

SELECT COUNT DISTINCT ( ?publishedReply )
WHERE {
  ?publishedReply rdf:type sioc:Post
  ?publishedReply foaf:maker personA .
  ?anyOtherPost sioc:has_reply ?publishedReply . }
```

Listing 5: Query for user's community involvement (R2)

```
SELECT COUNT DISTINCT ( ?replyingPerson )
WHERE {
  ?originalPost sioc:has_creator personA .
  ?originalPost sioc:has_reply ?reply .
  ?reply foaf:maker ?replyingPerson . }
```

The community involvement of a user (R2) is associated with the number of people from the community that have replied to his posts, where a higher number indicates relevance and visitors' involvement, as shown in Listing 5. This measure corresponds to the *indegree prestige* method in social network analysis [WF94, Sec. 5.3.2] applied to a graph formed between users via `sioc:has_reply` relationships.

SPARQL currently does not provide aggregate functions, therefore Listings 4 and 5 for illustration purposes use a pseudo-operator COUNT which we evaluate by counting a number of rows returned.

5 The Social SIOC Explorer prototype

Our prototype Social SIOC Explorer operates on the aggregated data described in section 3, extracts the social context as described in section 4 and allows users to browse and explore all disparate information in an integrated manner. The prototype can be used as a "social reader" to explore and subscribe to SIOC-enabled community sites such as weblogs, mailing lists, forums and IRC chats and includes the extracted social context of community information in the user interface.

Overview All SIOC content is integrated into a local RDF store and then displayed in various ways. The start page shows an overview of community sites and people. Users can decide to browse a particular forum, see the posts aggregated from all sites, or explore the profile of a particular user. Note that the posts here are not just weblogs posts but include posts from forums, IRC chats, mailing lists, etc. which are all described using the same SIOC RDF vocabulary.

Figure 2: An example post and aggregated replies

After selecting a particular forum, the user is presented with a list of posts in this forum in a reverse chronological order. As usual in feed readers, each post is summarised and can be opened to read the full content. An example of detailed view is shown in Figure 2, which includes metadata about the post, its complete text and all replies (which might have been posted outside of this particular weblog). Also, "lateral" social browsing is supported: clicking on an author or commenter jumps to this person's profile which includes all posts and replies written by this person across all forums; clicking on a topic shows all posts tagged with this topic, again across all forums. In contrast to ordinary feed readers, our lateral browsing works across all types of community forums: clicking on the user "Cloud" will not only show all his weblog posts, but also his contributions to mailing lists, IRC discussions, bulletin boards, etc.

Figure 3 shows an example of a person's description, including information from his FOAF profile such as his picture, homepage and interests, and also his extracted social context. The screenshot shows a summary of these relations, with more details and links to actual people in the social neighbourhood available by clicking the links. For this user, we see that he has written 338 posts and made 115 comments (R1), has received a total of 1454 replies, and that he knows 634 people through a shared discussion (R2) and knows 11 people directly (L1). All this information is extracted from multiple online community sites, e.g. replies and joint discussions that take place on another user's weblog are counted into this picture.

Development The prototype is built on the Ruby on Rails framework for Web application developmen and uses several components for consuming and processing Semantic Web data. One such component is ActiveRDF [ODG+07], which maps RDF data onto

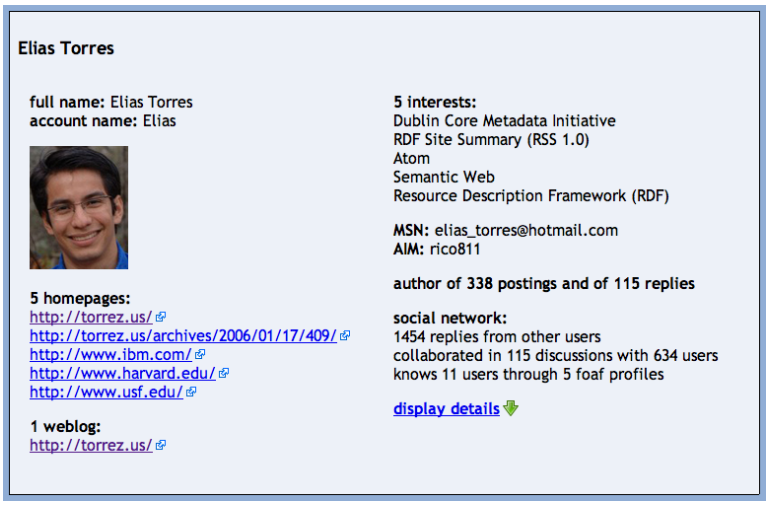


Figure 3: An example user profile with extracted social relations

programmatic objects. The second component is BrowseRDF [ODD06], a faceted browsing engine that enables navigation of large Semantic Web datasets without domain-specific navigation knowledge. The third component, written for a previous SIOC-based prototype, is a SIOC crawler that crawls, extracts, normalises, and integrates SIOC data from various community sites (which use different methods of exposing and linking to their SIOC data). The last component, added specifically for this use case, implements the social analysis algorithms described in section 4; it extracts social context information from the SIOC data and visualises this context in the user interface.

6 Related work

Flink [Mik05] is a web-based application for the extraction, analysis and visualisation of the social networks and research interests of Semantic Web researcher community. It uses electronic sources such as homepages, publications archives, and FOAF profiles as its data sources but does not consider online community sites such as weblogs. In comparison, SIOC provides us with rich and structured information from online community sites, including replies to posts, which enables us to perform this analysis.

Social network analysis methods [WF94] include calculation of metrics such as centrality, prestige, etc. The work presented here focuses on the prototype application for the exploration of online community sites and their networks. We therefore use rather simple and intiutive metrics such as indegree prestige. In future work more complex measures could be used to analyse the characteristics of these social networks, but that would require a larger dataset.

In terms of our navigation interface, related work can be found in other generic RDF

browsers that are also based on faceted navigation, such as Flamenco [YSLH03], mSpace [SWRS06]. Compared to these, our interface is more expressive in terms of navigation functionality [ODD06], but more importantly, these generic RDF browsers do not exploit the social aspect of the data.

Also, several graphical approaches exist for generic visual exploration of RDF graphs [FSvH04, FSvH02, Pie02] but these do not scale well for large graphs in terms of the user interface [FTH06].

7 Conclusion

We have presented an approach to extract social context from online social communities, and a prototype that exploits this information in the browsing process. By using the SIOC ontology we have access to high-quality data with rich structure, which we can directly analyse for implicit social relations. Relations between people can be derived from their online interactions, such as content that they create or reply to. We have introduced three levels of a user's social neighbourhood and two metrics of a user's social reputation, and have defined these as queries on the SIOC ontology. Finally, we have presented our prototype for browsing content from community sites based on these implicit social relations.

Acknowledgements This material is based upon works supported by the Science Foundation Ireland under Grants No. SFI/02/CE1/I131 and SFI/04/BR/CS0694. We gratefully acknowledge John Breslin's work on SIOC and feedback on this paper.

References

[BBP06] Uldis Bojārs, John G. Breslin, and Alexandre Passant. SIOC Browser – Towards a Richer Blog Browsing Experience. In Thomas N. Burg and Jan Schmidt, editors, *BlogTalks Reloaded: Social Software - Research & Cases*. Books on Demand GmbH, 2006.

[BHBD05] John G. Breslin, Andreas Harth, Uldis Bojārs, and Stefan Decker. Towards Semantically-Interlinked Online Communities. In *Proceedings of the European Semantic Web Conference (ESWC)*, 2005.

[BKvH02] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 54–68, 2002.

[FSvH02] Christiaan Fluit, Marta Sabou, and Frank van Harmelen. Ontology-based Information Visualization. In *Visualizing the Semantic Web*, pages 36–48. Springer-Verlag, 2002.

[FSvH04] Christiaan Fluit, Marta Sabou, and Frank van Harmelen. Supporting User Tasks through Visualisation of Light-weight Ontologies. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 415–434. Springer-Verlag, 2004.

[FTH06] Flavius Frasincar, Alexandru Telea, and Geert-Jan Houben. Adapting graph visualization techniques for the visualization of RDF data. In V. Geroimenko and C. Chen, editors, *Visualizing the Semantic Web*, chapter 9, pages 154–171. Springer-Verlag, second edition, 2006.

[HO07] Benjamin Heitmann and Eyal Oren. Leveraging existing Web frameworks for a SIOC explorer to browse online social communities. In *Proceedings of the ESWC Workshop on Scripting for the Semantic Web*, June 2007.

[KC97] Karin D. Knorr-Cetina. Sociality with Objects: Social Relations in Postsocial Knowledge Societies. *Theory, Culture and Society*, 14(4):1–30, 1997.

[KOM05] Atanas Kiryakov, Damyan Ognyanov, and Dimitar Manov. OWLIM – a Pragmatic Semantic Repository for OWL. In *Proceedings of the Conference on Web Information Systems Engineering (WISE) Workshops*, pages 182–192, 2005.

[Mik05] Peter Mika. Flink: Semantic Web technology for the extraction and analysis of social networks. *Journal of Web Semantics*, 3(2-3):211–223, 2005.

[ODD06] Eyal Oren, Renaud Delbru, and Stefan Decker. Extending faceted navigation for RDF data. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 559–572, November 2006.

[ODG+07] Eyal Oren, Renaud Delbru, Sebastian Gerke, Armin Haller, and Stefan Decker. ActiveRDF: Object-Oriented Semantic Web Programming. In *Proceedings of the International World-Wide Web Conference*, pages 817–823, May 2007.

[Pie02] Emmanuel Pietriga. *Environnements et Langages de Programmation Visuels pour le Traitement de Documents Structurés*. PhD thesis, Institut National Polytechnique de Grenoble, 2002.

[SWRS06] M. C. Schraefel, Max Wilson, Alistair Russell, and Daniel A. Smith. mSpace: Improving Information Access to Multimedia Domains with MultiModal Exploratory Search. *Communications of the ACM*, 49(4):47–49, 2006.

[WF94] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

[YSLH03] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 401–408, 2003.

RDF Support in the Virtuoso DBMS

Orri Erling
oerling@openlinksw.com

Ivan Mikhailov
imikhailov@openlinksw.com

Abstract: This paper discusses RDF related work in the context of OpenLink Virtuoso, a general purpose relational / federated database and applications platform. We discuss adapting a relational engine for native RDF support with dedicated data types, bitmap indexing and SQL optimizer techniques. We further discuss mapping existing relational data into RDF for SPARQL access without converting the data into physical triples. We present conclusions and metrics as well as a number of use cases, from DBpedia to bio informatics and collaborative web applications.

1 Introduction And Motivation

Virtuoso is a multi-protocol server providing ODBC/JDBC access to relational data stored either within Virtuoso itself or any combination of external relational databases. Besides catering for SQL clients, Virtuoso has a built-in HTTP server providing a DAV repository, SOAP and WS* protocol end points and dynamic web pages in a variety of scripting languages. Given this background and the present emergence of the semantic web, incorporating RDF functionality into the product is a logical next step. RDF data has been stored in relational databases since the inception of the model [1][8]. Performance considerations have however led to the development of custom RDF engines, e.g. RDF Gateway [7], Kowari [9] and others. Other vendors such as Oracle and OpenLink have opted for building a degree of native RDF support into an existing relational platform.

The RDF work on Virtuoso started by identifying problems of using an RDBMS for triple storage:

- Data Types: RDF is typed at run time and IRI's must be distinct from other data.
- Unknown data lengths large objects mix with small scalar values in a manner not known at query compile time.
- More permissive cast rules than in SQL.
- Difficulty of computing query cost. Normal SQL compiler statistics are not precise enough for optimizing a SPARQL query if if all data is in a single table.
- Efficient space utilization.
- Need to map existing relational data into RDF and join between RDF data and relational data.

We shall discuss our response to all these challenges in the course of this paper.

2 Triple Storage

Virtuoso's initial storage solution is fairly conventional: a single table of four columns holds one quad, i.e. triple plus graph per row. The columns are G for graph, P for predicate, S for subject and O for object. P, G and S are IRI ID's, for which we have a custom data type, distinguishable at run time from integer even though internally this is a 32 or 64 bit integer. The O column is of SQL type ANY, meaning any serializable SQL object, from scalar to array or user defined type instance. Indexing supports a lexicographic ordering of type ANY, meaning that with any two elements of compatible type, the order is that of the data type(s) in question with default collation.

Since O is a primary key part, we do not wish to have long O values repeated in the index. Hence O's of string type that are longer than 12 characters are assigned a unique ID and this ID is stored as the O of the quad table. For example Oracle [10] has chosen to give a unique ID to all distinct O's, regardless of type. We however store short O values inline and assign ID's only to long ones.

Generally, triples should be locatable given the S or a value of O. To this effect, the table is represented as two covering indices, G, S, P, O and O, G, P, S. Since both indices contain all columns, the table is wholly represented by these two indices and no other persistent data structure needs to be associated with it. Also there is never a need for a lookup of the main row from an index leaf.

Using the Wikipedia data set [12] as sample data, we find that the O is on the average 9 bytes long, making for an average index entry length of 6 (overhead) + 3 * 4 (G, S, P) + 9 (O) = 27 bytes per index entry, multiplied by 2 because of having two indices.

We note however that since S is the last key part of P, G, O, S and it is an integer-like scalar, we can represent it as a bitmap, one bitmap per distinct P, G, O. With the Wikipedia data set, this causes the space consumption of the second index to drop to about a third of the first index. We find that this index structure works well as long as the G is known. If the G is left unspecified, other representations have to be considered, as discussed below.

For example, answering queries like

```
graph <my-friends> {
    ?s sioc:knows
          <http://people.com/people#John> ,
          <http://people.com/people#Mary> }
```

the index structure allows the AND of the conditions to be calculated as a merge intersection of two sparse bitmaps.

The mapping between an IRI ID and the IRI is represented in two tables, one for the namespace prefixes and one for the local part of the name. The mapping between ID's of long O values and their full text is kept in a separate table, with the full text or its MD5 checksum as one key and the ID as primary key. This is similar to other implementations.

The type cast rules for comparison of data are different in SQL and SPARQL. SPARQL will silently fail where SQL signals an error. Virtuoso addresses this by providing a special

QUIETCAST query hint. This simplifies queries and frees the developer from writing complex cast expressions in SQL, also enhancing freedom for query optimization.

Other special SPARQL oriented accommodations include allowing blobs as sorting or distinct keys and supporting the IN predicate as a union of exact matches. The latter is useful for example with FROM NAMED, where a G is specified as one of many.

Compression. We have implemented compression at two levels. First, within each database page, we store distinct values only once and eliminate common prefixes of strings. Without key compression, we get 75 bytes per triple with a billion-triple LUBM data set (LUBM scale 8000). With compression, we get 35 bytes per triple. Thus, key compression doubles the working set while sacrificing no random access performance. A single triple out of a billion can be located in less than 5 microseconds with or without key compression. We observe a doubling of the working set when using 32 bit IRI ID's. The benefits of compression are still greater when using 64 bit IRI ID's.

When applying gzip to database pages, we see a typical compression to a third, even after key compression. This is understandable since indices are by nature repetitive, even if the repeating parts are shortened by key compression. Over 99% of 8K pages filled to 90% compress to less than 3K with gzip at default compression settings. This does not improve working set but saves disk. Detailed performance impact measurement is yet to be made.

Alternative Index Layouts. Most practical queries can be efficiently evaluated with the GSPO and OGPS indices. Some queries, such as ones that specify no graph are however next to impossible to evaluate with any large data set. Thus we have experimented with a table holding G, S, P, O as a dependent part of a row id and made 4 single column bitmap indices for G, S, P and O. In this way, no combination of criteria is penalized. However, performing the bitmap AND of 4 given parts to check for existence of a quad takes 2.5 times longer than the same check from a single 4 part index. The SQL optimizer can deal equally well with this index selection as any other, thus this layout may prove preferable in some use cases due to having no disastrous worst case.

3 SPARQL and SQL

Virtuoso offers SPARQL inside SQL, somewhat similarly to Oracleś RDF_MATCH table function. A SPARQL subquery or derived table is accepted either as a top level SQL statement of wherever a subquery or derived table is accepted. Thus SPARQL inherits all the aggregation and grouping functions of SQL, as well as any built-in or user defined functions. Another benefit of this is that all supported CLI's work directly with SPARQL, with no modifications. For example, one may write a PHP web page querying the triple store using the PHP to ODBC bridge. The SPARQL text simply has to be prefixed with the SPARQL keyword to distinguish it from SQL. A SPARQL end point for HTTP is equally available.

Internally, SPARQL is translated into SQL at the time of parsing the query. If all triples are in one table, the translation is straightforward, with union becoming a SQL union and optional becoming a left outer join. Since outer joins can be nested to arbitrary depths

inside derived tables in Virtuoso SQL, no special problems are encountered. The translator optimizes the data transferred between parts of the queries, so that variables needed only inside a derived table are not copied outside of it. If cardinalities are correctly predicted, the resulting execution plans are sensible. SPARQL features like construct and describe are implemented as user defined aggregates.

SQL Cost Model and RDF Queries. When all triples are stored in a single table, correct join order and join type decisions are difficult to make given only the table and column cardinalities for the RDF triple or quad table. Histograms for ranges of P, G, O, and S are also not useful. Our solution for this problem is to go look at the data itself when compiling the query. Since the SQL compiler is in the same process as the index hosting the data, this can be done whenever one or more leading key parts of an index are constants known at compile time. For example, in the previous example, of people knowing both John and Mary, the G, P and O are known for two triples. A single lookup in log(n) time retrieves the first part of the bitmap for

```
((G = <my-friends>) and (P = sioc:knows) and
  (O = <http://people.com/people#John>) )
```

The entire bitmap may span multiple pages in the index tree but reading the first bitts and knowing how many sibling leaves are referenced from upper levels of the tree with the same P, G, O allows calculating a ballpark cardinality for the P, G, O combination. The same estimate can be made either for the whole index, with no key part known, using a few random samples or any number of leading key parts given. While primarily motivated by RDF, the same technique works equally well with any relational index.

Basic RDF Inferencing. Much of basic T box inferencing such as subclasses and subproperties can be accomplished by query rewrite. We have integrated this capability directly in the Virtuoso SQL execution engine. With a query like

```
select ?person where { ?person a lubm:Professor }
```

we add an extra query graph node that will iterate over the subclasses of lubm:Professor and retrieve all persons that have any of these as rdf:type. When asking for the class of an IRI, we also return any superclasses. Thus the behavior is indistinguishable from having all the implied classes explicitly stored in the database.

For A box reasoning, Virtuoso has special support for owl:same-as. When either an O or S is compared with equality with an IRI, the IRI is expanded into the transitive closure of its same-as synonyms and each of these is tried in turn. Thus, when same-as expansion is enabled, the SQL query graph is transparently expanded to have an extra node joining each S or O to all synonyms of the given value. Thus,

```
select ?lat where { <Berlin> has_latitude ?lat }
```

will give the latitude of Berlin even if `<Berlin>` has no direct latitude but `geo:Berlin` does have a latitude and is declared to be `owl:same-as <Berlin>`.

The `owl:same-as` predicate of classes and properties can be handled in the T box through the same mechanism as subclasses and subproperties.

Data Manipulation. Virtuoso supports the SPARUL SPARQL extension, compatible with JENA [8]. Updates can be run either transactionally or with automatic commit after each modified triple. The latter mode is good for large batch updates since rollback information does not have to be kept and locking is minimal.

Full Text. All or selected string valued objects can be full text indexed. Queries like

```
select ?person from <people> where {
    ?person a person ; has_resume ?r .
    ?r bif:contains 'SQL and "semantic web"' }
```

will use the text index for resolving the pseudo-predicate `bif:contains`.

Aggregates. Basic SQL style aggregation is supported through queries like

```
select ?product sum (?value) from <sales> where {
    <ACME> has_order ?o .   ?o has_line ?ol .
    ?ol has_product ?product ; has_value ?value }
```

This returns the total value of orders by ACME grouped by product.

For SPARQL to compete with SQL for analytics, extensions such as returning expressions, quantified subqueries and the like are needed. The requirement for these is inevitable because financial data become available as RDF through the conversion of XBRL [13].

RDF Sponge. The Virtuoso SPARQL protocol end point can retrieve external resources for querying. Having retrieved an initial resource, it can automatically follow selected IRI's for retrieving additional resources. Several modes are possible: follow only selected links, such as `sioc:see_also` or try dereferencing any intermediate query results, for example. Resources thus retrieved are kept in their private graphs or they can be merged into a common graph. When they are kept in private graphs, HTTP caching headers are observed for caching, the local copy of a retrieved remote graph is usually kept for some limited time. The sponge procedure is extensible so it can extract RDF data from non-RDF resource via microformat or other sort of filter. This provides common tool to traverse sets of interlinked documents such as personal FOAFs that refer to each other.

4 Mapping Legacy Relational Data into RDF for SPARQL Access

RDF and ontologies form the remaining piece of the enterprise data integration puzzle. Many disparate legacy systems may be projected onto a common ontology using different rules, providing instant content for the semantic web. One example of this is OpenLink's ongoing project of mapping popular Web 2.0 applications such as Wordpress, Mediawiki, PHP BB and others onto SIOC through Virtuoso's RDF Views system.

The problem domain is well recognized, with work by D2RQ [2], SPASQL [5], DBLP [3]

among others. Virtuoso differs from these primarily in that it combines the mapping with native triple storage and may offer better distributed SQL query optimization through its long history as a SQL federated database.

In Virtuoso, an RDF mapping schema consists of declarations of one or more quad storages. The default quad storage declares that the system table RDF_QUAD consists of four columns (G, S, P and O) that contain fields of stored triples, using special formats that are suitable for arbitrary RDF nodes and literals. The storage can be extended as follows:

An IRI class defines that an SQL value or a tuple of SQL values can be converted into an IRI in a certain way, e.g., an IRI of a user account can be built from the user ID, a permalink of a blog post consists of host name, user name and post ID etc. A conversion of this sort may be declared as bijection so an IRI can be parsed into original SQL values. The compiler knows that an join on two IRIs calculated by same IRI class can be replaced with join on raw SQL values that can efficiently use native indexes of relational tables. It is also possible to declare one IRI class A as subClassOf other class B so the optimizer may simplify joins between values made by A and B if A is bijection.

Most of IRI classes are defined by format strings that is similar to one used in standard C sprintf function. Complex transformations may be specified by user-defined functions. In any case the definition may optionally provide a list of sprintf-style formats such that that any IRI made by the IRI class always match one of these formats. SPARQL optimizer pays attention to formats of created IRIs to eliminate joins between IRIs created by totally disjoint IRI classes. For two given sprintf format strings SPARQL optimizer can find a common subformat of these two or try to prove that no one IRI may match both formats.

```
prefix : <http://www.openlinksw.com/schemas/oplsioc#>
create iri class :user-iri "http://myhost/sys/users/%s" (
    in login_name varchar not null ) .
create iri class :blog-home "http://myhost/%s/home" (
    in blog_home varchar not null ) .
create iri class :permalink "http://myhost/%s/%d" (
    in blog_home varchar not null,
    in post_id integer not null ) .
make :user_iri subclass of :grantee_iri .
make :group_iri subclass of :grantee_iri .
```

IRI classes describe how to format SQL values but do not specify the origin of those values. This part of mapping declaration starts from a set of table aliases, somehow similar to FROM and WHERE clauses of an SQL SELECT statement. It lists some relational tables, assigns distinct aliases to them and provides logical conditions to join tables and to apply restrictions on table rows. When a SPARQL query should select relational data using some table aliases, the final SQL statement contains related table names and all conditions that refer to used aliases and does not refer to unused ones.

```
from SYS_USERS as user from SYS_BLOGS as blog
where (^{blog.}^.OWNER_ID = ^{user.}^.U_ID)
```

A quad map value describes how to compose one of four fields of an RDF quad. It may be an RDF literal constant, an IRI constant or an IRI class with a list of columns of table aliases where SQL values come from. A special case of a value class is the identity class, which is simply marked by table alias and a column name.

Four quad map values (for `G`, `S`, `P` and `O`) form quad map pattern that specify how the column values of table aliases are combined into an RDF quad. The quad map pattern can also specify restrictions on column values that can be mapped. E.g., the following pattern will map a join of `SYS_USERS` and `SYS_BLOGS` into quads with `:homepage` predicate.

```
graph <http://myhost/users>
subject :user-iri (user.U_ID)
predicate :homepage
object :blog-home (blog.HOMEPAGE)
where (not ^{user.}^.U_ACCOUNT_DISABLED) .
```

Quad map patterns may be organized into trees. A quad map pattern may act as a root of a subtree if it specifies only some quad map values but not all four; other patterns of subtree specify the rest. A typical use case is a root pattern that specifies only the graph value whereas every subordinate pattern specifies `S`, `P` and `O` and inherits `G` from root, as below:

```
graph <http://myhost/users> option (exclusive) {
    :user-iri (user.U_ID)
      rdf:type foaf:Person ;
      foaf:name user.U_FULL_NAME ;
      foaf:mbox user.U_E_MAIL ;
      foaf:homepage :blog-home (blog.HOMEPAGE) . }
```

This grouping is not only a syntax sugar. In this example, `exclusive` option of the root pattern permits the SPARQL optimizer to assume that the RDF graph contains only triples mapped by four subordinates.

A tree of a quad map pattern and all its subordinates is called “RDF view” if the “root” pattern of the tree is not a subordinate of any other quad map pattern.

Quad map patterns can be named; these names are used to alter mapping rules without destroying and re-creating the whole mapping schema.

The top-level items of the data mapping metadata are quad storages. A quad storage is a named list of RDF views. A SPARQL query will be executed using only quad patterns of views of the specified quad storage.

Declarations of IRI classes, value classes and quad patterns are shared between all quad storages of an RDF mapping schema but any quad storage contains only a subset of all available quad patterns. Two quad storages are always defined: a default that is used if no storage is specified in the SPARQL query and a storage that refers to single table of physical quads.

The RDF mapping schema is stored as triples in a dedicated graph in the `RDF_QUAD` table so it can be queried via SPARQL or exported for debug/backup purposes.

5 Applications and Benchmarks

As of this writing, July 2007, the native Virtuoso triple store is available as a part of the Virtuoso open source and commercial offerings. The RDF Views system is part of the offering but access to remote relational data is limited to the commercial version.

Virtuoso has been used for hosting many of the data sets in the Linking Open Data Project [14], including Dbpedia [15], Musicbrainz [16], Geonames [17], PingTheSemanticWeb [18] and others. The largest databases are in the single billions of triples.

The life sciences demonstration at WWW 2007 [19] by Science Commons was made on Virtuoso, running a 350 million triple database combining diverse biomedical data sets.

Web 2.0 Applications. We can presently host many popular web 2.0 applications in Virtuoso, with Virtuoso serving as the DBMS and also optionally as the PHP web server.

We have presently mapped PHP BB, Mediawiki and Drupal into SIOC with RDF Views.

OpenLink Data Spaces (ODS). ODS is a web applications suite consisting of a blog, wiki, social network, news reader and other components. All the data managed by these applications is available for SPARQL querying as SIOC instance data. This is done through maintaining a copy of the relevant data as physical triples as well as through accessing the relational tables themselves via RDF Views.

LUBM Benchmark. Virtuoso has been benchmarked with loading the LUBM data set. At a scale of 8000 universities, amounting to 1068 million triples, the data without key compression size is 75G all inclusive and the load takes 23h 45m on a machine with 8G memory and two 2GHz dual core Intel Xeon processors. The loading takes advantage of SMP and parallel IO to disks. With key compression the data size drops to half.

Loading speed for the LUBM data as RDFXML is 23000 triples per second if all data fits in memory and 10000 triples per second with one disk out of 6 busy at all times. Loading speed for data in the Turtle syntax is up to 38000 triples per second.

6 Future Directions

Clustering. Going from the billions into the tens and hundreds of billions of triples, the insert and query load needs to be shared among a number of machines. We are presently implementing clustering support to this effect. The present clustering scheme can work in a shared nothing setting, partitioning individual indices by hash of selected key parts. The clustering support is a generic RDBMS feature and will work equally well with all RDF index layouts. We have considered Oracle RAC[20]-style cache fusion but have opted for hash partitioning in order to have a more predictable number of intra cluster messages and for greater ease in combining messages.

The key observation is that an interprocess round-trip in a single SMP box takes 50 microseconds and finding a triple takes under five. Supposing a very fast interconnect and a cluster of two machines, the break-even point after which cluster parallelism wins over

message delays is when a single message carries 10 lookups. Thus, batching operations is key to getting any benefit from a cluster and having a fixed partition scheme makes this much more practical than a shared disk/cache fusion architecture such as Oracle RAC.

Updating Relational Data by SPARUL Statements. In many cases, an RDF view contains quad map patterns that maps all columns of some table into triples in such a way that sets of triples made from different columns are "obviously" pairwise disjoint and invoked IRI classes are bijections. E.g., quad map patterns for RDF property tables usually satisfy these restrictions because different columns are for different predicates and column values are used as object literals unchanged. We are presently extending SPARUL compiler and run-time in order to make such RDF views updatable.

The translation of a given RDF graph into SQL data manipulation statement begins with extracting all SQL values from all calculatable fields of triples and partitioning the graph into groups of triples, one group per one distinct extracted primary key of some source table. Some triples may become members of more than one group, e.g., a triple may specify relation between two table rows. After integrity check, every group is converted into one insert or delete statement.

The partitioning of N triples requires $O(N \ln N)$ operations and keeps data in memory so it's bad for big dump/restore operations but pretty effecient for transactions of limited size, like individual bookeeping records, personal FOAF files etc.

7 Conclusion

Experience with Virtuoso has encountered most of the known issues of RDF storage and has shown that without overwhelmingly large modifications, a relational engine can be molded to efficiently support RDF. This has also resulted in generic features which benefit the relational side of the product as well. The reason is that RDF makes relatively greater demands on a DBMS than relational applications dealing with the same data.

Applications such as `www.pingthesemanticweb.com` and ODS for social networks and on-line collaboration have proven to be good test-beds for the technology. Optimizing queries produced by expanding SPARQL into unions of multiple storage scenarios has proven to be complex and needing more work. Still, it remains possible to write short queries which are almost impossible to efficiently evaluate, especially if they join between data that may come from many alternate relational sources. In complex EDI scenarios, some restrictions on possible query types may have to be introduced.

The community working on RDF storage will need to work on interoperability, standard benchmarks and SPARQL end point self-description. Through the community addressing these issues, the users may better assess which tool is best suited for which scale of problem and vendors may provide support for query and storage federation in an Internet-scale multi-vendor semantic web infrastructure.

Further details on the SQL to RDF mapping and triple storage performance issues are found in separate papers on the http://virtuoso.openlinksw.com site.

Literatur

[1] Beckett, D.: Redland RDF Application Framework. http://librdf.org/

[2] Bizer, C., Cyganiak, R., Garbers, J., Maresch, O.: D2RQ: Treating Non-RDF Databases as Virtual RDF Graphs. http://sites.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/

[3] Chen, H., Wang, Y., Wang, H. et al.: Towards a Semantic Web of Relational Databases: a Practical Semantic Toolkit and an In-Use Case from Traditional Chinese Medicine. http://iswc2006.semanticweb.org/items/Chen2006kx.pdf

[4] Guo, Y., Pan, Z., Heflin, J.: LUBM: A Benchmark for OWL Knowledge Base Systems. Journal of Web Semantics 3(2), 2005, pp158–182.
Available via http://www.websemanticsjournal.org/ps/pub/2005-16

[5] Prudhommeaux E.: SPASQL: SPARQL Support In MySQL.
http://xtech06.usefulinc.com/schedule/paper/156

[6] 3store, an RDF triple store. http://sourceforge.net/projects/threestore

[7] Intellidimension RDF Gateway. http://www.intellidimension.com

[8] Jena Semantic Web Framework. http://jena.sourceforge.net/

[9] Northrop Grumman Corporation: Kowari Metastore. http://www.kowari.org/

[10] Oracle Semantic Technologies Center.
http://www.oracle.com/technology/tech/semantic_technologies/index.html

[11] Semantically-Interlinked Online Communities. http://sioc-project.org/

[12] Wikipedia3: A Conversion of the English Wikipedia into RDF.
http://labs.systemone.at/wikipedia3

[13] Extensible Business Reporting Language (XBRL) 2.1.
http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2006-12-18.rtf

[14] Bizer C., Heath T., Ayers D., Raimond Y.: Interlinking Open Data on the Web. 4th European Semantic Web Conference.
http://www.eswc2007.org/pdf/demo-pdf/LinkingOpenData.pdf

[15] Sören Auer, Jens Lehmann: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content 4th European Semantic Web Conference.
http://www.informatik.uni-leipzig.de/ auer/publication/ExtractingSemantics.pdf

[16] About MusicBrainz. http://musicbrainz.org/doc/AboutMusicBrainz

[17] About Geonames. http://www.geonames.org/about.html

[18] Ping The Semantic Web. http://pingthesemanticweb.com/about.php

[19] Alan Ruttenberg: Harnessing the Semantic Web to Answer Scientific Questions. 16th International World Wide Web Conference.
http://www.w3.org/2007/Talks/www2007-AnsweringScientificQuestions-Ruttenberg.pdf

[20] Oracle Real Application Clusters. http://www.oracle.com/database/rac_home.html

Implementing SPARQL Support for Relational Databases and Possible Enhancements

Christian Weiske, Sören Auer
Universität Leipzig
cweiske@cweiske.de, auer@informatik.uni-leipzig.de

Abstract: In order to make the Semantic Web real we need the infrastructure to store, query and update information adhering to the RDF paradigm. Such infrastructure can be developed from scratch or benefit from developments and experiences made in other science & technology realms such as within the database domain. For querying RDF data the World Wide Web Consortium released a Working Draft for the SPARQL query language. A large portion of the Web is meanwhile driven by server-side Web applications. PHP is the scripting language most widely used for Web applications. In this paper we present our PHP implementation of the SPARQL standard directly interacting with an underlying database system. The approach is based on the rationale of pushing as much work into the RDBMS as possible in order to profit most from the query optimization techniques developed for relational database systems. The article includes an evaluation of the performance and standard compliance, surveys limitations we discovered when using SPARQL for the implementation of real-life Semantic Web applications and suggests extensions to the SPARQL standard in order to solve these obstacles.

1 Introduction

In order to make the Semantic Web real we need the infrastructure to store, query and update information adhering to the RDF paradigm. Such infrastructure can be developed from scratch or benefit from developments and experiences made in other science & technology realms such as within the database domain. Database systems, for example, have reached a very high degree of maturity with respect to data storage and querying: The SQL standard allows to formulate queries regarding almost all possible nuances of relational algebra. Database systems include sophisticated query optimizers finding optimal data access plans independent from specific query phrasings. Hence, building on top of such full-grown technologies will enable Semantic Web developers to benefit greatly from advances of database systems.

For querying RDF data the World Wide Web Consortium released a Working Draft for the SPARQL query language[PS06]. A large portion of the Web is meanwhile driven by server-side Web applications. PHP is the scripting language most widely used for Web applications[1]. In this paper we present our PHP implementation of the SPARQL standard directly interacting with an underlying database system and suggest enhancements to the

[1] `http://www.tiobe.com/tpci.htm`

SPARQL standard, which are motivated by real-life applications of our implementation such as OntoWiki [ADR06] and DBpedia [ABL+07].

The approach is based on the rationale of pushing as much work into the RDBMS as possible in order to profit most from the query optimization techniques developed for database systems. This article covers implementation details of the new SPARQL engine, evaluates its performance and coverage of the proposed SPARQL standard, surveys limitations we discovered when using SPARQL for the implementation of semantic web applications, as well as suggests extensions to the SPARQL standard in order to solve these obstacles.

2 A SPARQL Engine for PHP

The interface between the programming language (such as PHP in our case) and the database query language (SPARQL) is an application programming interface (API). A few PHP-based open-source RDF APIs are available, and RAP (RDF API for PHP, [Biz04]) is one of the most mature one amongst them. One of the limitations of RAP was its SPARQL engine. It is built to work on any RDF model that can be loaded into memory. Using SPARQL to query a database required to load the complete database into memory and execute the SPARQL query on it. While this works well with some dozens of RDF triples, it can not be used for databases with millions of triple data - main memory is one limitation, execution time another one (code implemented in a scripting language such as PHP is about 50 times slower[2] than pure C implementations of the same code). Our goal was to create a SPARQL engine that pushes as much work as possible to the database server, using PHP only as translator between SPARQL and SQL [Ins89] and integrating seamlessly into the existing RAP infrastructure.

2.1 Database vs. Main Memory Implementation

RAP's old SparqlEngine implementation works on RDF stores whose data are loaded into main memory. Querying 100 million triples stored in a SQL database causes the whole data to be loaded into the server's RAM - often multiple times, depending on the number of triple patterns the query uses.

Other than to the old implementation, SparqlEngineDb creates SQL queries from the internal SPARQL query representation and lets the database server execute these queries. Query results are converted into the desired result format as the last step.

Unlike the old SparqlEngine implementation, our new SparqlEngineDb pushes all hard and time consuming operations into the database server. Since databases are highly optimized for selecting, filtering, ordering and joining data we can expect a significant speedup using this approach.

[2] `http://shootout.alioth.debian.org/gp4/benchmark.php?test=all&lang=php&lang2=gcc`

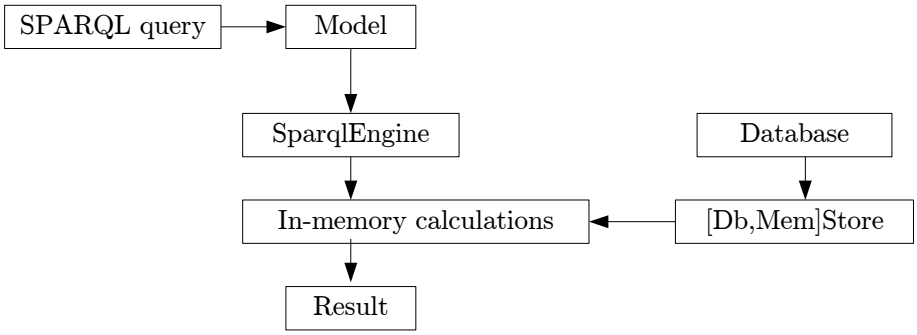


Figure 1: SparqlEngine data flow

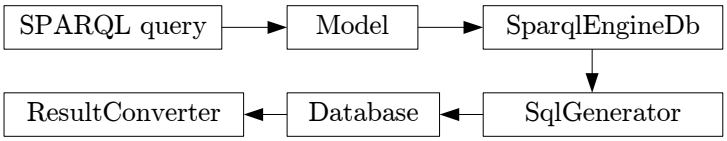


Figure 2: SparqlEngineDb data flow.

Another concept introduced in SparqlEngineDb are result renderers. The old engine had one default result type, PHP arrays. A parameter was added to the query method to allow returning XML data, and the engine had an extra method `writeQueryResultAsHtmlTable` that added formatting capabilities to the model class implementation. Beside mixing of view and controller classes, the existing system was in no way extensible or customizable. Core classes needed to be changed to add new output formats.

`SparqlEngineDb` (and now also the old memory based `SparqlEngine`) support `ResultRenderer` - pluggable drivers that convert raw database results into the desired output format - be it SPARQL Query Results XML Format, [BB06], the JavaScript Object Notation [3] JSON for web applications or an own application specific renderer. The system allows great flexibility without ever needing to change internal classes.

2.2 Generating SQL

The main job SparqlEngineDb performes is generating one or several SQL queries from a SPARQL query's object representation after parsing it.

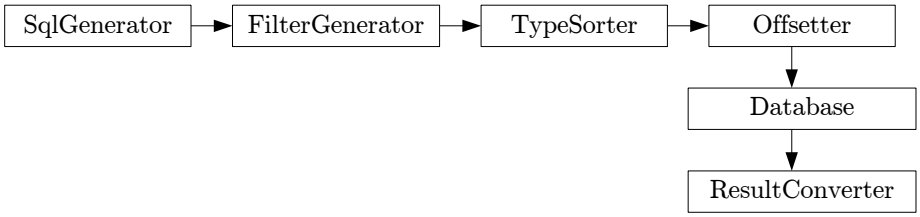
SQL queries are created specifically for RAP's database schema. This unnormalized

[3] `http://www.json.org/`

schema collects all triple data in a single "statements" table. The table structure is visualized below:

Column name	Data type
modelID	integer
subject	varchar
subject_is	varchar(1)
predicate	varchar
object	blob
object_is	varchar(1)
l_language	varchar
l_datatype	varchar

The SQL query generation involves the following four steps before the query is complete and can be sent to the database:



After generating SQL joins for each individual triple pattern, the filter generator creates suitable SQL `WHERE` clauses from SPARQL filter expressions.

Since RAP stores RDF triples in the database in an unnormalized way, boolean `TRUE` values for example may have multiple representations in SQL: `T`, `TRUE` and `1`. The same applies to all datatypes, such as for example `xsd:dateTime` values, as the following example illustrates:
`"2004-12-31T18:01:00-05:00"^^<xsd:dateTime>` [4] and
`"2004-12-31T20:01:00-03:00"^^<xsd:dateTime>`

Both literals represent the same date and time, but are physically stored differently. Queries on such unnormalized data are more complex since all possible different value combinations have to be checked.

In RAP, literal values of all data types are stored as BLOBs in the object column of the statements table. Using a simple `ORDER BY object` is not possible, because for example numerical values such as `10` and `2` would be returned in the wrong order. Hence, type casting is necessary, but quite hard and cumbersome to implement correctly considering the unnormalized representation of the data.

Several actions needed to be undertaken in order to deal with these problems. Certain equations in SPARQL's `FILTER` clauses are expanded to several `WHERE` clauses in SQL

[4] We use the namespaceprefix `xsd` for http://www.w3.org/2001/XMLSchema#

and concatenated with `OR`. SPARQL's `ORDER BY` clauses need special attention since different data types need to be casted differently. Here, we determine first a list of all data types in the result set and execute one query for each one of the found data types. The final result set is constructed by combining the individual results by means of the SQL `UNION` operator.

Sending multiple queries to the SQL server requires custom handling of `LIMIT` and `OFFSET` constrains. Hence, the Offsetter determines the number of result rows for each single query and calculates which queries to execute at all, and which parts of it to return.

All these inconveniences make `SparqlEngineDb` slower than it would be with a normalized database layout. Future versions of RAP should alternatively provide support for storing triples in normalized form.

3 Evaluation

Our SPARQL implementation was evaluated along three dimensions: we compared its performance with other state of the art engines, evaluate its compliance with the SPARQL standard and review some examples of its usage in real applications.

3.1 Performance Comparison

We used the Lehigh University Benchmark tool suite [GPH05] to generate RDF data for the query speed evaluation. Queries have been tested on databases with sizes ranging from five to 200.000 triples. The competitors were RAP's old memory based *SparqlEngine*, our new *SparqlEngineDb*, *Jena SDB*, *ARC*, *Redland/librdf* and *Virtuoso*. Two groups of SPARQL queries have been selected for evaluation[5]. Queries of first group focus each on a different SPARQL feature and were designed to be independent of RDF data structure and types, i.e. can be used with any concrete dataset. The second group contains three queries which are specific to the Lehigh Benchmark RDF data and aim at evaluating the runtime of complex queries.

Using the queries, we seek answers to the following questions:

1. How does the implementation behave on small vs. large databases?
2. Which features do the individual implementations support?
3. Are filter clauses optimized in some way?
4. How does the engine behave on complex queries?
5. Which implementation is the fastest?

[5]The evaluation queries can be found at:
`http://ontowiki.net/files/SPARQL-PerformanceEvaluationDetails.pdf`

We found out that some engines do not fully support the SPARQL specification and did not deliver results for some queries. The results of the performance comparison show that Virtuoso is by far the fastest SPARQL implementation available. Our new SparqlEngineDb implementation performs second best on average in all tests with database sizes from 5 to 200.000 triples. Only SparqlEngineDb and Jena were able to provide results for all tested queries. Redland did not execute the `UNION` query while ARC failed on nearly half of the tests (cf. Figure 3).

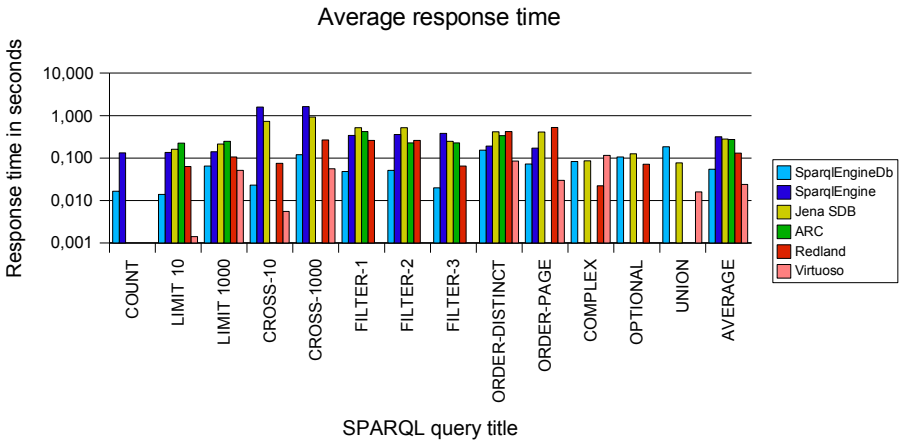


Figure 3: Average response times for different queries.

Based on the results of the performance evaluation, we see the following improvement areas for SparqlEngineDb:

- Multiple sort variables cause the engine to return results multiple times slower than using just a single variable to sort results.
- Optional patterns are fast on small databases, but get significantly slower if the size increases. The increase of computing time is much slower or even constant with other engines.
- Limit handling in UNION queries needs to be improved. Currently, limits are applied only after the full result set has been calculated.

3.2 SPARQL Standard Compliance

A great amount of work at SparqlEngineDb has been done in order to comply with the SPARQL specification. RAP contains a comprehensive unit test suite that is used to ensure all aspects of the package are working correctly. The SPARQL test suite in RAP includes

the test cases created by the RDF Data Access Working Group[6] as well as tests created by the programmers, reproducing former bugs. The total amount of SPARQL test cases in RAP is 151; SparqlEngineDb passes 148 of them.

SparqlEngineDb supports all SPARQL features except named graphs and sorting dates with different timezones. Named graph support could not be implemented due to missing support for named graphs in RAP's current database layout. Sorting time data works correctly on dates in the same time zone only. The reason for this is the lack of time zone support in SQL in string to date conversion functionality.

The performance evaluation tests of different SPARQL engines have shown that SparqlEngineDb and Jena SBD are the only engines to fully support all tested queries.

3.3 Current usage

The development of the new SPARQL engine was driven by the needs of real application projects. During development, support for SparqlEngineDb usage has been added to numerous applications. The following list shows some of them:

- Ontowiki[ADR06], is a semantic wiki engine working completely on RDF data. OntoWiki uses SPARQL to access the underlying databases.
- LDAP2SPARQL[7] provides an LDAP interface to SPARQL endpoints. It allows using LDAP capable clients to retrieve data from RDF databases.
- Vakantieland[8] is a knowledge base for tourist information in the Netherlands and retrieves all data from an RDF database.

While working on these application projects we noted a number of shortcomings which we hope might be tackled in future versions of SPARQL. We summarize the most important ones in the next section.

4 Possible SPARQL Enhancements

SPARQL allows to query RDF data easily. It supports filtering data by simple conditions like regular expressions, data type checks and boolean combinations of them. In comparison to SQL, however, SPARQL's language feature set is quite limited. Features currently not supported by SPARQL are:

- Operating on distinct models

[6] http://www.w3.org/2001/sw/DataAccess/
[7] http://aksw.org/Projects/LDAP/Backend
[8] http://www.vakantieland.nl

- Using operators in `SELECT` and `CONSTRUCT` patterns
- Mathematic functions
- Aggregates and grouping functionality
- Prepared statements
- Data manipulation and transaction control
- Query concatenation
- Variable renaming
- Access control

Using SPARQL today often still means retrieving raw selected data from the SPARQL endpoint. Consecutively, the client needs to re-examine all fetched data as well as filter and convert them into the required format. This produces unnecessary overhead that could be avoided if SPARQL would support data transformation. Pushing more 'work' into the SPARQL server also means that applications are more light-weight and portable across different systems and programming languages. From our point of view, SPARQL should become for RDF graphs what SQL already is for relational databases. SPARQL is still a W3C working draft, thus it is possible that missing features might be added before the standard is finalized. Some ideas for SPARQL enhancements are explained in more detail in the remainder of this section.

4.1 Operating on Distinct Models

Following the "SQL for graphs" approach, the following missing feature becomes apparent: SPARQL is only able to operate on a single "database" - RDF triples belonging to a single model. It is often desirable that a database server supports multiple stores that keep data independent of each other. Most current SPARQL server implementations include support to handle distinct models.

While named graphs try to address this problem, it is often favorable to have fully distinct sets of RDF data that do not influence each other. Accidentally or intentionally omitting the graph name causes a query to be executed on the whole model. Separating models of e.g. different users from each other allows better access control. Further, data loss is restricted to a single model when accidentally executing a delete operation.

With the growth of SPARQL end point implementations, the number of non-standardized, different APIs addressing this problem is growing. Before finalizing the SPARQL specification, a solution for this problem should be provided in order to prevent incompatible solutions between server software implementations. The selection of a specific model in SPARQL can for example be easily realized with an additional keyword `USE`:

```
USE <http://example.com/addressbook>
```

It should be possible to obtain a list of all models available on the server. The syntax of such a query should adhere to the normal SPARQL standard. For example, a special system model (similar to the information schema of relational databases) could be provided and queried as follows for a list of available models:

```
USE <http://example.com/sysmodel>
PREFIX sysmodel: <http://example.com/sysmodel-schema/>
SELECT ?model WHERE { ?x rdf:type sysmodel:model }
```

4.2 SPARQL as a Calculator

The SPARQL working draft supports ”extensible value testing” by allowing vendors to add custom functions to their SPARQL implementation. An example is given that shows an `aGeo:distance` function calculating the distance between two locations.

Such functionality can be either provided by adding proprietary functions to the server, or by extending the SPARQL specification to support further base functionality. Standard functionality like trigonometric functions or exponents are regularly needed and can thus be added to the standard feature set of SPARQL. Such a step prevents incompatibilities between implementations and allows applications to be portable between SPARQL endpoints.

Supporting elementary mathematical calculations opens SPARQL to a wide range of possible SPARQL queries and use cases. Further, it drastically reduces the need for client side data filtering and proprietary extensions in SPARQL server implementations. The working draft's example function `aGeo:distance`, for example, could be replaced by using power and a square root:

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
SELECT ?neighbor WHERE {
  ?a geo:placeName "Grenoble".
  ?a geo:lat ?ax. ?a geo:long ?ay.
  ?b geo:placeName ?neighbor.
  ?b geo:lat ?bx. ?b geo:long ?by.
  FILTER ( sqrt(pwr(?ax-?bx,2)+pwr(?ay-?by,2)) < 10 ).
}
```

4.3 Extended Operator Usage

SPARQL is limited in a way that only full values of subjects, predicates and objects can be returned from queries. Modifying values of selected variables through the query allows pushing tasks into the database server and reduces the need for data manipulation on client side.

Another often needed feature is to find out which languages are used in conjunction with literals in the data store. SPARQL currently does not provide any way to get a list of all used languages or datatypes. This problem can be solved by allowing operators on variable selection clauses:

```
SELECT DISTINCT datatype(?o) WHERE  { ?s ?p ?o }
```

By adding such functionality, renaming result variables gets important:

```
SELECT DISTINCT substr(str(?o), 0, 10) AS ?name
  WHERE  { ?s ?p ?o }
```

4.4 Grouping and Aggregates

Grouping and aggregate functionality is a useful feature currently missing in SPARQL. Virtuoso already implements aggregate functions by means of implicit grouping[9]. Implicit grouping as Virtuoso provides introduces some problems:

- Queries are hard to understand because variable selection and cgroup creation is mixed.
- It is not possible to select variables without using them to create a group.
- Grouping is only available globally, not for single graph patterns.
- Operators cannot be used on group variables.

Virtuoso's approach provides basic grouping functionality but should be extended to prevent these problems. Readability difficulties and the negative effects of implicit grouping can be prevented by introducing a new clause similar to SQL's `GROUP BY`.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name, count(?mbox) as ?count
WHERE { ?x foaf:name ?name.
        ?x foaf:mbox ?mbox. }
GROUP BY ?name
```

It has to be defined at which position in a SPARQL query a `GROUP BY` clause should be allowed. For that it has to be decided whether it is desirable to be able to apply grouping to individual graph patterns. The `ORDER BY` clause is also only allowed on a global level, although it could make sense for "subqueries" like optional clauses.

[9] `http://docs.openlinksw.com/virtuoso/rdfsparqlaggregate.html`

4.5 Prepared Statements

In SQL, prepared statements serve several purposes:

- Queries do need to be parsed only once and can be executed multiple times without further parsing overhead.
- Query execution plans need to be calculated only once.
- For web applications, they prevent SQL injection attacks, since there is no need to escape values.
- The amount of data transmitted between client and SQL server is reduced.

Prepared statements can speed up repetitive queries of the same form a great deal and enhance security. By supporting prepared statements, SPARQL would benefit in the same way. We propose the following prepared statement syntax rules:

- Placeholders are defined using a normal SPARQL variable name but using two consecutive question marks as prefix.
- SPARQL implementations need two new API methods, namely `prepare` and `execute`.

Using named placeholders (as Oracle does for SQL[10]) has several advantages over position-based anonymous placeholders (e.g. MySQL [11]):

- The user does not need to know the exact order of parameters the query requires.
- Repeated usage of the same variable requires only one data instance to be provided and transmitted to the server. This saves memory and bandwidth.

The preparing and execution of a SPARQL statement with placeholders in PHP would for example would look as follows:

```
$query = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?mbox
WHERE { ?x foaf:name ??name. ?x foaf:mbox ?mbox. }";

$statement = $model->prepare($query);

$data1 = $statement->execute(array('name'=>'John Doe'));
$data2 = $statement->execute(array('name'=>'Jane Doe'));
```

[10] http://download-east.oracle.com/docs/cd/B19306_01/appdev.102/b14250/oci05bnd.htm

[11] http://dev.mysql.com/doc/refman/5.0/en/sqlps.html

5 Conclusion and Outlook

We presented our SPARQL implementation SparqlEngineDb, which for the first time enables the largest portion of Web application (implemented in PHP) to base on semantic technologies. When compared to other SPARQL implementations SparqlEngineDb is mostly faster, only preceded by Virtuoso, which technically follows a very similar approach but is implemented solely in C.

Use-case of SparqlEngineDb showed that for implementing real-world applications the use of SPARQL is still cumbersome and lack some crucial features. Our vision is that SPARQL eventually will support a similar comprehensive querying of RDF graphs as SQL allows of relational databases. As a first step in that direction we suggested several enhancements to the SPARQL standard, which could be implemented in a quite canonical downward compatible way.

With a growing usage of SPARQL as the standard to query RDF data, the Semantic Web will gain more interoperability. By continuously evolving the SPARQL standard, it will become easier to create sophisticated applications that operate on data stored in different server implementations.

References

[ABL+07] Sören Auer, Chris Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of ISWC 2007, November 11-15, Busan, Korea.*, 2007.

[ADR06] Sören Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006.

[BB06] Dave Beckett and Jeen Broekstra. SPARQL Query Results XML Format. W3c candidate recommendation, World Wide Web Consortium (W3C), April 2006.

[Biz04] Chris Bizer. RAP (RDF API for PHP). Website, November 2004. http://www.wiwiss.fu-berlin.de/suhl/bizer/rdfapi/.

[GPH05] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *J. Web Sem*, 3(2-3):158–182, 2005.

[Ins89] American National Standards Institute. *Information Systems -Database Language - SQL*. ANSI X3.135-1992. American National Standards Institute, 1430 Broadway, New York, NY 10018, USA, 1989.

[PS06] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF (Working Draft). W3c working draft, World Wide Web Consortium (W3C), 2006.

Collaborative Metadata for Geographic Information

Patrick Maué

pajoma@uni-muenster.de

Abstract: Retrieval of web-based geographic information (GI) for spatial decision-making processes can benefit from emerging semantic technologies. Ontology-supported metadata, collaboratively created by a social network of spatially-aware users, ensures efficient and precise discovery of maps published with the help of catalogs. The creation and maintenance of the metadata is driven by two forces: on the one hand experienced catalogers who ensure consistency and quality, on the other hand catalog users who continuously adapt and extend the metadata. We discuss the requirements for a catalog for GI which is able to capture the semantics emerging within a community and apply the results to the registered metadata.

1 Introduction

One of the definitions found in [Pic95] treats geographic information systems (GIS) as a set of technologies used to collect, manipulate and represent GI. A common application of GIS are spatial decision-making tasks, for example in the context of urban planning. Research on participatory GIS (PGIS) aims to open up GIS to involve the public into such processes [Ren06]. PGIS can include a variety of approaches to make "GIS and other spatial decision-making tools available and accessible to all those with a stake in official decisions" [SS05]. The public can be involved several times during the process, in the following we will focus on GI retrieval, evaluation and maintenance.

GI for PGIS is either created directly by the participating organizations or retrieved from external sources like the World Wide Web. Relying on web-based GI is particular: everyone interested is able to join the decision-making process [Tul07]. In addition, it ensures access to GI without the need of expensive, high-level equipment [LME+02] commonly required for local GIS. The increasing availability of web mapping tools either used to browse maps (e.g. Google Maps, available at http://maps.google.com) or to create maps (e.g. Open Street Map, available at http://www.openstreetmap.org) is the reason that people are getting used to basic map interaction techniques formerly reserved for GIS specialists. And the ongoing transition of the user's role from content consumer to content producer [Kel05] accounts for the increasing availability of community-based GI. But it remains mostly unused for PGIS due to the lack of standardized APIs or sophisticated discovery tools. The problem of missing standards has been addressed by the Open Geospatial Consortium (OGC). Spatial data infrastructures (SDI) are proposed for the distributed supply of GI in the web, and how SDI and PGIS can be combined has been investigated in [KWR05].

The lack of sophisticated discovery tools has several reasons, one is the problem of varying possibilities to provide GI. Web services, conformal to the standards by the OGC, are the usual way to access GI in a SDI. But community-based GI is also often provided as downloadable files encoded in, for example, GeoRSS or Google's KML. Catalogs, which are the common approach for the discovery of GI available within a SDI, should support the different solutions for GI retrieval. Metadata needs to be published to make GI findable using a catalog. Such metadata can include information about the nature of the data, how to access the data or the service, who has created the data, and more. Catalogs also provide interfaces to query the repository of registered records and retrieve the matching entries. Searching GI requires to extend the common notion of keyword-based queries: spatial filters can be added to restrict the search results to a specific region of the world.

In this paper we discuss metadata registered in a catalog used to publish and search GI. Users send queries based on given vocabulary and a reasoner infers which entries in the repository match the query statements. But relying on sophisticated query processing algorithms alone can not provide better results, if the registered metadata lacks quantity and quality. Insufficient metadata is the main reason if search results suffer from low precision, poor rate of relevant entries, inconsistent search results, and many more [CJL+06]. Searching, for example, the web for OGC-conformal Web Map Services (WMS) does return a long result list. But only a minority of the GI provider make use of the extensive capabilities suggested by the WMS standard. Sometimes not even a title or a short abstract are provided. This problem is common for web-based GI and makes searching suitable data a tedious and unsatisfactory task.

In the following section we further investigate the nature of metadata and discuss why elaborated metadata is crucial for the usability of GI. Making the data provider or professional catalogers responsible for the creation and maintenance of the metadata is the common approach and its drawbacks are discussed afterwards. At the end of this section, the idea of a collaborative creation of the metadata is introduced. Letting the community contribute to the existing metadata can help to ensure more comprehensive and consistent descriptions of the underlying data. Before we come to a conclusion, we try to put things together in the fourth section by applying the results from the preceding sections to the already introduced catalog for GI.

2 Metadata based on Ontologies

GI comprises two elements: data and a description of the data. Unlike metadata of information usually exchanged in the web, e.g. images or videos, an elaborated description of spatial data is crucial for a successful integration. A GIS depends on it to display the complex data correctly. GI can have a varying spatial dimension, can have a temporal extent, and is very variable concerning the thematic features. The thematic properties usually represent observed phenomena, e.g. the amount of precipitation at a certain location. In this case, metadata should include not only the unit of measurement, but also a description or the accuracy of the measurement process.

Metadata can be modeled as the triple $M_d = (FP_d, NFP_d, U)$ for a particular data set d. The non functional properties NFP are used to describe the externalities of the data and can include an open list of features a data provider might want to add. It might contain a title, a short description, information about licensing issues, usage fees, and more. The information about the data provider is represented by U. FP_d is a set of partly mandatory functional properties for d, which has to include at least details about accessing the data, for example the URL linking to the file containing the data. GI provided with the help of a web service requires more information, for example the protocol used to invoke it. In addition of such primary functional aspects, the structure and the semantics of the data itself should be described as well to achieve seamless integration. A description of the data structure ensures correct display of, for example, a map showing last month's precipitation. Describing the semantics is more challenging and subject of ongoing research efforts. But only if the meaning of the data has been captured in its metadata, we can ensure a sound interpretation of the GI. Some value hidden within the attributes is supposed to represent the measurement of precipitation, but this information is only reliable if this information is explicitly stated. Semantic annotations address this problem and are further explained below.

One important aspect influencing the usability of GI is its findability, which depends on elaborated metadata. It has to be extensible and adaptable to changes of the underlying data [SL06]. Schema-based metadata, for example encoded in XML, can fulfill this requirement only partially. Metadata based on ontologies is more flexible, enables modularization [DHSW02] and helps to prevent semantic conflicts. The latter exist due to semantic heterogeneities like homonyms and synonyms or due to different levels of detail of the conceptualization. Having, for example, a concept labeled *rain* would not match queries asking for *precipitation*. Ontology alignment in the form of semantic annotations provides a solution to relate concepts describing the nature of the data to concepts denoting real world phenomena [KFM07]. The latter are stored within domain ontologies, which also provide the vocabulary used for building semantic queries. A user selects the appropriate concept from the domain ontology, for example *precipitation*, and sends the semantic query to the catalog. Reasoning algorithms are able to infer that maps providing information about *rain* are relevant for this query, if their metadata has been annotated with the selected domain concept.

But even with a semantically-enhanced catalog, the user still lacks the possibility of evaluating the GI if the metadata contains only little information. Information asymmetry between the data provider and the data requester can make GI useless if quality is the crucial factor. In the context of spatial decision-making processes, the data requester will rather choose to buy data from companies known for high-quality data than rely on GI available in the web if he is not able to evaluate it. Next to the requirement to prevent semantic conflicts we therefore also have to ensure that metadata is comprehensive and contains sufficient information for the evaluation of the described data. Metadata creation is usually in the responsibility of the data creator or professional catalogers; the drawbacks of this approach are discussed in the next section.

3 Creating metadata

Publishing GI using a catalog relies on the data provider or a professional cataloger to create its metadata. A typical place facing the same challenge of searching and classifying information is the library, and *Library Research* has addressed many of the information retrieval problems which now have become an issue for GI retrieval. In libraries the book is the subject of interest for students, and professional catalogers are responsible for indexing the books to make them findable using a library catalog.

> ”The cataloger must envisage the needs of the reader, endeavoring in every way to make it a simple process for him to find books. He should, like the librarian, adopt a neutral stand between the reader and his books, giving emphasis to what the author intended to describe rather than to his own views”

This quote by Margaret Mann [Man30] elucidates the cataloger’s problem. He has to anticipate not only which keywords the creator would assign, but also which keywords the user would use to search the book in question. Library Research has always faced the problem of the information gap between the searching students and indexing cataloger due to differing backgrounds [Hey99]. Indexing is a subjective process which accounts for the problem that ”one person’s view on an item in a retrieval system my hinder retrieval by others” [Bat98]. Furthermore, to ensure consistency and simplified maintenance, the controlled vocabulary used for the classification has to be minimized. This results inevitably in incomplete metadata. Incorporating all the keywords students could possibly use to search the book in question is not an option.

A problem usually more apparent in the web is the requirement for a continuous adjustment of the metadata. As the underlying data, whether GI or online texts, is changing, the describing view on it has to change as well [APF06]. Member fluctuation renders web communities dynamic: users with additional knowledge providing new information might join. Metadata needs to be continuously updated within such environments. Letting professional catalogers create and maintain the catalogs ensures high-quality metadata, but can also be expensive and time-consuming [Mat04]. With the growing importance of the classification of resources made available through the web, this problem got even more urgent. At last, the responsibility of the data providers to create the metadata during the registration process is often the reason for low-quality metadata. The professional cataloger has an interest to keep the used vocabulary consistent and the quality of the metadata on a certain level. The data provider on the other hand has no motivation to provide elaborated and consistent information if he gains no benefit. This problem, also known as moral hazard [Del06], can have a significant impact on the usability of data.

One option to address the mentioned problems is to involve the user in the categorization. In [HR97], users have been made responsible for indexing items, in this case images. The user terms have then been aggregated to a generalized view on the resources, called the public index. This process is also called distributed classification and realizes a bottom-up consensus view of the world [Spe07] . Several examples exist where the user is involved in classification, mostly by letting them assign tags to the items in question. Citeulike (http://www.citeulike.org) implements a catalog of research articles, LibraryThing

(http://www.librarything.org) lets users create personal libraries. Both solutions aggregate personal tags to create a public index for tag-based discovery. Exploiting the collective intelligence hidden within the community of catalog users to create and maintain more sophisticated metadata is the preferred option to avoid the drawbacks of the usual top-down approach. A collaborative approach helps to incorporate the varying user contexts in the metadata, is less costly in terms of money and time, makes metadata very adaptable to changes from outside, and is a way to ensure comprehensive metadata in general.

Contributions can come from GIS users applying the retrieved data in their own applications. They can, for example, add new non-functional properties or apply semantic annotations to enrich the functional data descriptions. Contributions can also be produced by semi-automatic mining algorithms, which examine the structure of the data to capture the semantics hidden within the maps. Geographical features in the database may imply information about the topological relations of the represented features [KL05]. Or the co-occurrence of keywords or words within the title or description of two maps might indicate similarity [SRT05]. Or user profiles can be analyzed and compared to reason about the degree of affiliation [LMD06]. All refer to a kind of semantics ”that is implicit from the patterns in data and that is not represented explicitly in any strict machine processable syntax” [SRT05]. All can be subsumed as implicit semantics which are, together with the powerful (soft) and the formal semantics [SRT05], an approach to include the community in the ontology building and maturing process [BSW+07].

Implicit semantics can be captured in different ways, one approach used for the catalog is discussed in the next session. The derived information can then be used as input for updating the powerful (soft) semantics encoded in the non-functional part of the metadata ontologies. They are called soft because they can be fuzzy, contain inconsistencies and uncertainties within the relations of the concepts. Formal semantics on the other hand are explicit, have to be consistent and decidable. They are represented by the functional part of the metadata, the semantic annotations and the domain ontologies. The fluctuation of the community and changing views on the world result in changing implicit semantics, and therefore powerful (soft) semantics as well. Capturing such emergent semantics [AMO+04] requires continuous synchronization between the implicit semantics hidden within the system and the interactions between the users and the powerful (soft) semantics.

Neither the bottom-up, the collaborative approach, nor the top-down approach with the data provider alone as creator of the metadata are able to succeed without the other [Kli03]. The data provider has much more information about the creation process and the nature of the data and is therefore responsible for most of the metadata. The top-down approach ensures consistency and therefore interoperability. But one imposed world view would never fit all participating parties, a bottom-up approach has to be part of the metadata creation process as well. Relying on a bottom-up approach does not necessarily result in higher rates of recall [GT06]. Incorporating many world views without having a controlling instance leads to inconsistent and fuzzy ontologies, which drives them unusable for many applications [Pet06]. In the next section we suggest a strategy for a GI catalog with a simple approach to capture the implicit semantics to exploit the collective intelligence and ensure high-quality metadata.

4 Bridging the information gap

Up until now, this paper has introduced the concept of ontology-supported metadata and explained two methods for its creation: either in form of a top-down approach by letting the data provider create the data descriptions or as bottom-up process letting the community build the metadata in collaboration. We also argued that both approaches should actually play a complementary role. In this section we propose a possible solution to bridge the gap between the implicit semantics coming from the user interacting with the catalog and the formal semantics used for the discovery.

But there is no straightforward way to exploit the implicit semantics. In the context of our catalog, we focus on the actions a user can perform with the catalog. Contributing actions have an effect on the implicit semantics, and should therefore have an impact on the metadata ontologies capturing the powerful (soft) semantics as well. The similarity between two metadata records is used as indicator for changing implicit semantics. Similarity has a threefold use. Searching the catalog can use it as indicator to assess the relevance of the search results. Browsing through the registered GI can be supported by similarities, equal to URLs linking to related documents in the web. And similar records can be used as clue for changing formal semantics, for example finding the appropriate domain concepts for the semantic annotations.

To capture the implicit semantics, we could either keep a log of all modifications within a certain time span and calculate the changes of the similarities all in one. The alternative which will be used for the catalog is to let the effects of every modification directly change the related values. All possible operations a user can perform on the catalog are classified, depending on their effect on similarities and the affiliation of the participating users. The actions, both implicit and explicit, can be confirming or contributing. A user might, for example, load specific GI into a local GIS system and confirm that a service providing the data works as expected. As next step, he might contribute to the metadata by rating the GI. *Tagging* and *Relevance Feedback* are two typical examples of an explicit and implicit action, their meaning and effects are described in the following.

Tagging A folksonomy is the compilation of terms members of the catalog community choose to tag specific GI with. One dataset providing coffee bars in Paris can have the tags {*coffeehouse, paris, coffee, bar*}. This short list already shows that tags are subjective and imprecise [GT06]. Only the first tag does really describe the nature of the represented location. Some like the the label ”paris” give only contextual information. And tags are noisy and error-prone, only the combination ”coffee bar” is correct, but bar only has multiple meanings in English. Tags can be used either to measure similarity of the information provided by two datasets, or act as input for semantic annotations.

Relevance Feedback Querying the catalog using keywords returns usually multiple results. To improve the overall precision of the discovery, relevance feedback mechanisms can be incorporated [BYRN99]. From the result set the searching user can select a set or records that appear to be relevant for the query. Future similar searches can exploit this information and put the formerly records marked as relevant to the

top of the result. Information collected from the relevance feedbacks can be exploited, like the tags, to modify the similarity between the marked entries.

Metadata for a particular dataset d is defined as the triple $M_d = (FP_d, NFP_d, U)$. Two metadata entries are similar, if some of their content overlaps, for example if one or more keywords are equal or if the two described services provide information about the same area. Metadata and the similarities are formalized as weighted, directed graph $G_M = \{M, S, W_S\}$. M is a set of nodes representing the metadata entries registered at the catalog. $S \subset M \times M$ is the set of similarities between two metadata entries, with the similarity $W_S : S \rightarrow [0, 1]$. W_S is an unreliable value and should be regarded only as an indicator for the similarity. Two entries might be similar because the described data covers the same geographic space, for example the area of a city. Else, the data have nothing in common: one provides a street network, the other precipitation values. When used for assessing relevant search results, similarity is an ad-hoc value depending on the requested information. The suggested solution used for the catalog is the permanent storage of queries in the catalog. The graph G_M does not only include the metadata records M_d , but also the query records M_r. A query record has the same structure as M_d, but is describing desired GI, not one actually existing in the repository. Having both, metadata and query records, in the graph allows for similarities between predefined search queries and existing database records. Searching the catalog is now different: the user can either select on of the existing query records, e.g. describing street data, adapt it or create a new one. Adding query records to the repository results in an evolving catalog which (a) is able to adapt to the user requirements and (b) allows for context-aware inquiries. This is a novel feature which helps to improve the acceptance of catalogs for the discovery of GI.

Due to the fact that user contributions can affect the popularity of the registered GI, the user and user relations have to be modeled in a network as well. Similar to the metadata and its similarities, the social network can be modeled as a graph $G_U = \{U, A, W_A\}$. Here, the nodes are the users and the edges are the degree of affiliation between the users. A user's rating of the quality of another user's GI does have an impact on the overall quality rating of the map. And maps with higher quality are considered more valuable, and gain a better position in searching results. Having the acquaintance modeled in the system can prevent that users knowing each other can take advantage of the collaborative approach. A user's rating will have less impact on the overall quality of a map, if he has a relation to the user who created the map. Taking the degree of affiliation between the catalog users into account to assess the reputation and the effects of an operations will help to avoid fraud. This novel approach is therefore important to ensure the trust between the users and the credibility of the underlying application, the catalog.

5 Conclusion

Defining the effects of implicit and explicit actions on the powerful (soft) semantics, namely the similarities of metadata records, can help to capture some of the emergent semantics. The impact on the formal semantics on the other hand have been barely dis-

cussed. Formal semantics on the domain level require consistent and sound ontologies, which can usually only be ensured by skilled catalogers. Semantic annotations are part of the formal semantics, and a mistake here can render the described map unusable. To avoid the risk of unusable domain ontologies, future work will include an investigation how we can assess the experience of the catalog users. Every user is able to search the catalog and influence the implicit semantics, but only few experienced should be able to change the formal semantics. We propose reputation, which is built upon the history of past interactions and can be used as indicator for the experience of the user. A reputation value of a particular user can increase or decrease during the time depending on the interaction with the catalog. We will assemble different kinds of user contributions and classify them in terms of impact on the findability and their dependency on reputation accordingly. The modeling approach for the metadata and query records, the domain ontologies, and the reputation of catalog users has been barely discussed in this paper. Further research is required here as well to address open issues like the formalization, scalability and robustness of the model.

This paper was focusing on a catalog for GI. But the approach for a collaborative creation of metadata is neither restricted to GI nor to catalogs. Relying on a catalog can have serious drawbacks like single point of failure, lack of scalability, and also issues of trust. With the advent of distributed user identity systems like OpenID (see http://www.openid.net), we also need to implement methods to decouple a user's reputation from the actual catalog and make it available to other platforms, for example catalogs organized in a Peer-to-Peer network.

We have proposed a collaborative approach for creating and modifying metadata for GI. The metadata model is extensible and allows for semantic-based discovery. Relying on cataloger alone for the creation of metadata makes the descriptions not flexible, comprehensive and reliable enough. A collaborative approach helps to avoid these problems, but can only play a complementary role to ensure consistency. Incorporating the user can help to close the gap between the implicit, informal and dynamic semantics hidden within the system on the one side and the explicit, formal and stable domain knowledge on the other. A novel contribution of the discussed approach is the use of reputation for this, and in particular how this reputation is built depending on the affiliation to other users. In the context of catalogs for GI used for PPGIS, the idea of incorporating the user to increase the usability is novel as well. But we believe that all web-based applications which are used for the publication and discovery of all kinds of information will be able to improve the findability and therefore also the usefulness of the managed information, if they incorporate the proposed approach.

6 Acknowledgments

This research has been supported by the EU-IST Project No. FP6-26514 (SWING) . I would also like to thank the reviewers and Prof. Dr. Werner Kuhn for their valuable comments.

References

[AMO+04] Karl Aberer, Philippe C. Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand S. Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, Olga De Troyer, Thomas Risse, Monica Scannapieco, Fèlix Saltor, Luca de Santis, Stefano Spaccapietra, Steffen Staab, and Rudi Studer. Emergent Semantics Principles and Issues. In *Proceedings of Database Systems for Advances Applications (DASFAA 2004)*, pages 25–38. Springer, March 2004.

[APF06] Grigoris Antoniou, Dimitris Plexousakis, and Giorgos Flouris. Evolving Ontology Evolution. In *Proceedings of Theory and Practice of Computer Science (SOFSEM 2006)*, pages 14–29. Springer Berlin / Heidelberg, 2006.

[Bat98] Marcia J. Bates. Indexing and access for digital libraries and the internet: Human, database, and domain factors. *Journal of the American Society for Information Science*, 49(13):1185–1205, December 1998.

[BSW+07] Simone Braun, Andreas Schmidt, Andreas Walter, Gabor Nagypal, and Valentin Zacharias. Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering. In *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at the 16th International World Wide Web Conference (WWW 07), Banff, Canada*, 2007.

[BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[CJL+06] Rafael T. Calasanz, Jos, Javier Lacasta, Javier N. Iso, Pedro, and Javier. On the Problem of Identifying the Quality of Geographic Metadata. In *Proceedings of Research and Advanced Technology for Digital Libraries (ECDL 2006), Alicante, Spain*, pages 232–243, September 2006.

[Del06] Chrysanthos Dellarocas. *Handbook on Information Systems and Economics*, chapter Reputation Mechanisms. Elsevier Publishing, 2006.

[DHSW02] Erik Duval, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4), April 2002.

[GT06] Marieke Guy and Emma Tonkin. Folksonomies: Tidying up Tags? *D-Lib Magazine*, 12(1), 2006.

[Hey99] Francis Heylighen. Collective Intelligence and its Implementation on the Web: Algorithms to Develop a Collective Mental Map. *Computational & Mathematical Organization Theory*, 5(3):253–280, October 1999.

[HR97] Rob Hidderley and Pauline Rafferty. Democratic indexing: An approach to the retrieval of fiction. *Information Services & Use*, 17(2/3):101–109, 1997.

[Kel05] Kevin Kelly. We Are the Web, August 2005.

[KFM07] Eva Klien, Daniel I. Fitzner, and Patrick Maué. Baseline for Registering and Annotating Geodata in a Semantic Web Service Framework. In *Proceedings of 10th International Conference on Geographic Information Science (AGILE 2007)*, 2007.

[KL05] Eva Klien and Michael Lutz. The Role of Spatial Relations in Automating the Semantic Annotation of Geodata. In A. Cohn and D. Mark, editors, *Proceedings of the Conference on Spatial Information Theory (COSIT'05)*, volume LNCS 3693, pages 133–148, Ellicottville, NY, USA, 2005. Springer.

[Kli03] Ralf Klischewski. Top down or bottom up? How to Establish a Common Ground for Semantic Interoperability within e-Government Communities. In R. Traunmller and M. Palmirani, editors, *Proceedings of 1st International Workshop on E-Government at ICAIL 2003. Bologna*, pages 17–26. Gedit edizioni, 2003.

[KWR05] Carsten Keßler, Marion Wilde, and Martin Raubal. Using SDI-based Public Participation for Conflict Resolution. In *Proceedings of the 11th EC-GI & GIS Workshop, Alghero, Sardinia*, 2005.

[LMD06] Hugo Liu, Pattie Maes, and Glorianna Davenport. Unraveling the Taste Fabric of Social Networks. *International Journal on Semantic Web and Information Systems*, 2(1):42–71, 2006.

[LME+02] Helga Leitner, Robert McMaster, Sarah Elwood, Susanna McMaster, and Eric Sheppard. Models for making GIS available to community organizations: Dimensions of difference and appropriateness. In Craig, Trevor M. Harris, and Daniel Weiner, editors, *Community Participation and Geographical Information Systems*, pages 37–53. CRC, April 2002.

[Man30] Margaret Mann. *Introduction to cataloging and classification of books*. American Library Association, Chicago, 1930.

[Mat04] Adam Mathes. *Folksonomies: Cooperative classification and communication through shared metadata*. PhD thesis, University of Illinois, Urbana-Champaign, December 2004.

[Pet06] Elaine Peterson. Beneath the Metadata - Some Philosophical Problems with Folksonomy. *D-Lib Magazine*, 12(11), November 2006.

[Pic95] John Pickles. Representations in an Electronic Age: Geography, GIS, and Democracy. In John Pickles, editor, *Ground Truth: The Social Implications of Geographic Information Systems*, pages 1–30. New York: The Guilford Press, 1995.

[Ren06] Sieber Renee. Public Participation Geographic Information Systems: A Literature Review and Framework. *Annals of the Association of American Geographers*, 96(3):491–507, September 2006.

[SL06] Nadine Schuurman and Agnieszka Leszczynski. Ontology-Based Metadata. *Transactions in GIS*, 10(5):709–726, November 2006.

[Spe07] Edith Speller. Collaborative tagging, folksonomies, distributed classification or ethnoclassification: a literature review. *Library Student Journal*, February 2007.

[SRT05] Amit P. Sheth, Cartic Ramakrishnan, and Christopher Thomas. Semantics for the Semantic Web: The Implicit, the Formal and the Powerful. *Int. J. Semantic Web Inf. Syst.*, 1(1):1–18, 2005.

[SS05] Marc Schlossberg and Elliot Shuford. Delineating ”Public” and ”Participation” in PPGIS. *URISA Journal*, 16(2):15–26, 2005.

[Tul07] David L. Tulloch. Many, many maps: Empowerment and online participatory mapping. *First Monday*, 12(2), February 2007.

Mapping Cognitive Models to Social Semantic Spaces – Collaborative Development of Project Ontologies

Thomas Riechert[1], Steffen Lohmann[2]

[1]University of Leipzig
Department of Computer Science
Johannisgasse 26, 04009 Leipzig, Germany
riechert@informatik.uni-leipzig.de

[2]University of Duisburg-Essen
Dep. of Informatics and Applied Cognitive Science
Lotharstrasse 65, 47057 Duisburg, Germany
lohmann@interactivesystems.info

Abstract: In this paper, we present an approach that applies concepts from the areas of Social Software and Semantic Web to application development. We start with a short introduction into Semantic Based Requirements Engineering. Then, we present an ontology for capturing requirements and related information. Based on this ontology, we describe how stakeholders are enabled to directly participate in requirements elicitation and to collaboratively create a project ontology. Finally, we report about the application of the presented approach in a use case from the e-government domain.

1 Introduction

The goal of software development is to build applications that meet the users' needs. Involving all relevant stakeholders is therefore a crucial part of the development process. However, every stakeholder has its own view and expectations on the application to be developed. The creation of a shared understanding and terminology is often very challenging. Requirements engineering methods, ranging from heavyweight [Roy87] to agile [Bec00], aim to face these problems. Their major task is to model the application on an abstract level and from different points of view. However, these models cannot be directly created by the stakeholders. In general, the stakeholder statements are translated in the vocabulary of the software designers before aggregated in the respective models.

Semantic Web and Social Software open up new opportunities to cope with these difficulties. Within the SoftWiki research project[1] we develop a web based collaborative environment that fosters direct stakeholder participation in early stages of requirements engineering. The SoftWiki philosophy follows the notion of the Social Semantic Web: Participation should be as easy as possible and semantically structured at the same time.

[1] http://softwiki.de

This paper presents two main results of our research: a semantic model for requirements engineering and a method to directly involve stakeholders in the collaborative development of project ontologies.

2 Semantic-based Requirements Engineering

Ontologies attracted significant attention in software engineering recently. Developments such as the W3C's Ontology Driven Architecture (ODA)[2] or the OMG's Ontology Definition Metamodel (ODM)[3] testify the growing interest for ontology-based approaches in software engineering and related disciplines. The advantages of ontology-based approaches compared to conventional ones, however, often remain unclear. To motivate the use of ontologies, one can argue with the vision of the Semantic Web and its need for a comprehensive ontological grounding, but this would presumably not convince many companies to invest in ontology construction. Another argument is the facilitation of automation in subsequent development phases when basing the requirements on an ontological structure.

Besides these arguments, we regard an underlying ontological structure as highly valuable for integrating distributed stakeholders in the requirements engineering process and for building consensus among them. A semantically unambiguous, well-structured means to represent shared views and prevent (costly) misunderstandings is crucial – in this regard, ontologies have their strength. Ontologies may serve as a suitable *Interlingua* between the stakeholders and system designers. Requirements engineering would then consist partly in ontological engineering, but with the particular and challenging constraint that the ontological structure is understandable and usable by various stakeholders with restricted or no backgrounds in ontology construction.

3 SoftWiki Ontology for Requirements Engineering

In order to semantically support the requirements engineering process we developed the SoftWiki Ontology for Requirements Engineering (SWORE) in accordance with standards of the requirements engineering community [Poh07, HPW98, Lam01]. Figure 1 visualizes the core of SWORE. Central to our approach are the classes *Stakeholder* and *Abstract Requirement* as well as the properties *details* and *defines*. Abstract requirements have the subclasses *Goal*, *Scenario* and *Requirement*, each of which are defined by stakeholders and can be detailed by other abstract requirements. This enables the specification of requirements at different levels of granularity. We emphasize the collaborative aspects of requirements engineering by integrating discussions amongst the stakeholders and voting (with the criteria of agreement and importance) in the model. This documentation is often relevant for future decisions in the requirements engineering process.

[2] http://www.w3.org/2001/sw/BestPractices/SE/ODA/060211/
[3] http://www.omg.org/ontology/

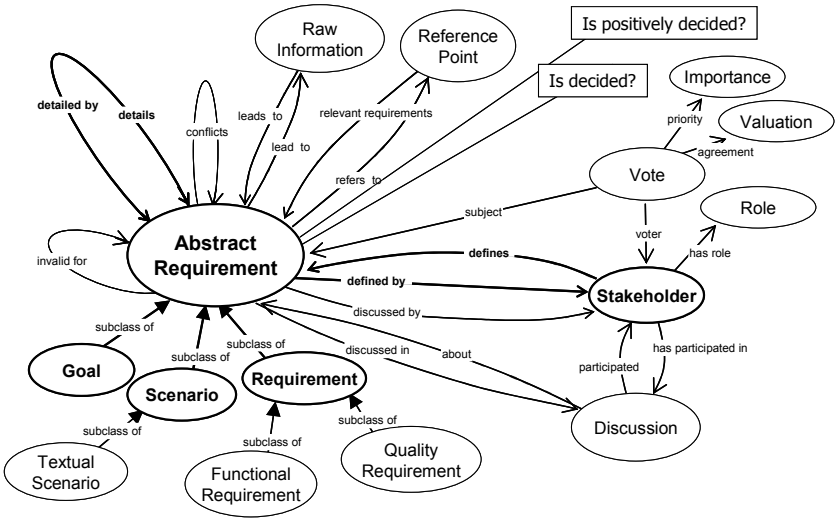


Figure 1: Visualisation of the SWORE core

The use of SWORE is visualised in Figure 2. A requirements engineering knowledge base contains instantiations ("is a") of the SWORE concepts. Furthermore, every abstract requirement refers to one or more concepts of the project ontology via its reference point.

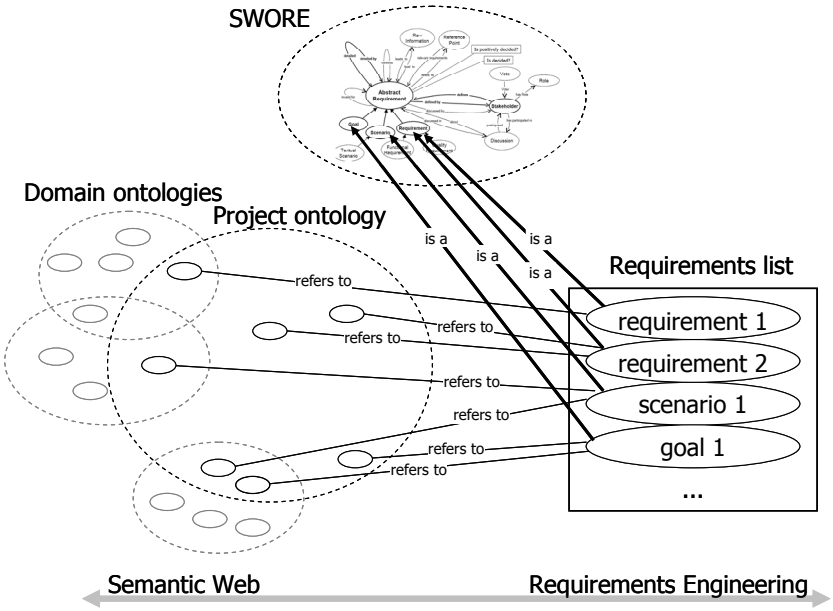


Figure 2: SWORE use and interlinking with the project ontology

The following example of a functional requirement illustrates the interlinking with the project ontology.

"The e-mail application should load messages from several email providers."

The stakeholder that formulates this requirement may assume that the model should contain concepts such as *message* and *email provider*, while another stakeholder may assume the model contains a concept such as *data transfer* (cp. Figure 3).

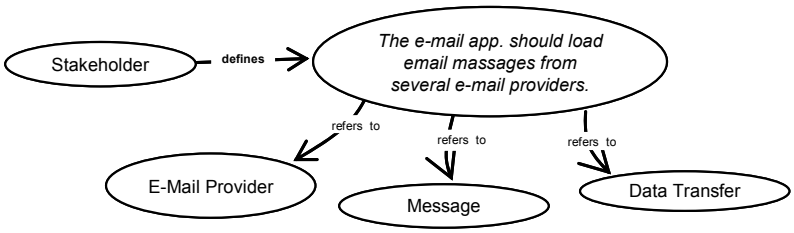


Figure 3: Requirement and links to instances of the class *Reference Point* equal to *owl:Thing*

The referenced concepts form the basis of the project ontology. Some of the concepts may also be parts of general domain knowledge that is already available in the Web[4]. This way, the requirements are enriched with further semantics. The project and domain ontologies provide useful semantic contexts to the stakeholders. This is illustrated by the following example of a requirement:

"The email application should support the most common data transfer protocols."

Stakeholders may refer this requirement to the concepts *data transfer* or *POP3 protocol*. Accessing an ontology that contains knowledge on the topic of email technologies, further concepts may be retrieved that are semantically related to the referenced concepts such as *IMAP protocol*.

4 Collaborative Development of Project Ontologies

The described method of collaboratively assigning reference points to requirements can also be regarded as a semantically enriched form of social tagging. As illustrated in Figure 4, additional relationships emerge in this tagging process (cp. [Sa06, Ma06]). On the one hand, relations between requirements evolve as two or more requirements get tagged with the same reference points (semantic tags). On the other hand, stakeholders become linked when using the same reference points (tag based relations) or tagging the same requirements (requirement based relations).

[4] This domain knowledge could have been created with the help of semantic extraction algorithms such as it is the case in the SoftWiki-related DBpedia project where formal knowledge is extracted from Wikipedia contents [AL07].

The reference points provide a valuable overview on the requirements and enable their exploration from different angles. Following the model of "perspective making and perspective taking" [BT95], the individual is faced with other stakeholders' views on the planned application that are expressed by the reference points they assigned to the requirements. New insights or categorization schemes may result that positively effect the requirements engineering process. Moreover, reference points used by the stakeholders may indicate their expertise with respect to a specific topic („I tag, therefore I know" [JS06]). Consequently, these concepts can serve as a starting point for fruitful discussions about the application in mind and might foster knowledge exchange and requirements refinement. Finally, the collaboratively created reference points provide a consolidated conceptual basis that can facilitate further software development. Particularly with regard to software design, some of the reference points may indicate components, technologies, or features that should be part of the planned application.

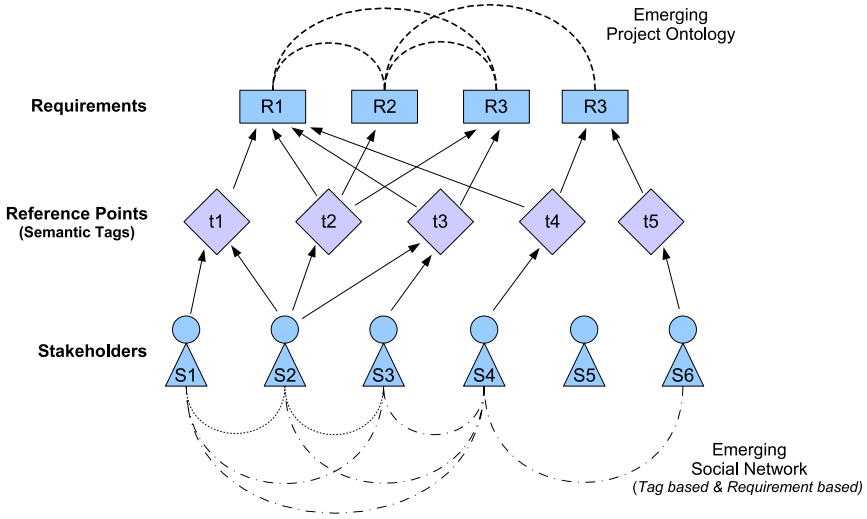


Figure 4: Collaborative semantic tagging of requirements by stakeholders

5 E-Government Use Case

An application and evaluation of the presented approach is currently performed in an e-government use case. The situation under consideration is characterized by a multiplicity of applications that have been developed and are in use by individual administration offices. In order to facilitate automated data exchange between different community administrations and to provide comprehensive web-based citizen services there are endeavours to homogenize and better integrate these individual applications on a local, regional and federal level. This is not just a technical problem, since different community administrations established different processes and information structures and hence have different requirements regarding information integration and exchange.

In order to enable the different stakeholders to collaboratively work on requirements for standardization, information and process integration we employed the described approach of Semantic Based Requirements Engineering. We adopted the semantic collaboration platform OntoWiki[5] [ADR06] for this purpose. OntoWiki enables a distributed user community to develop ontologies and collect instance data adhering to those ontologies. It aims at making browsing of knowledge bases and user contribution as easy as possible.

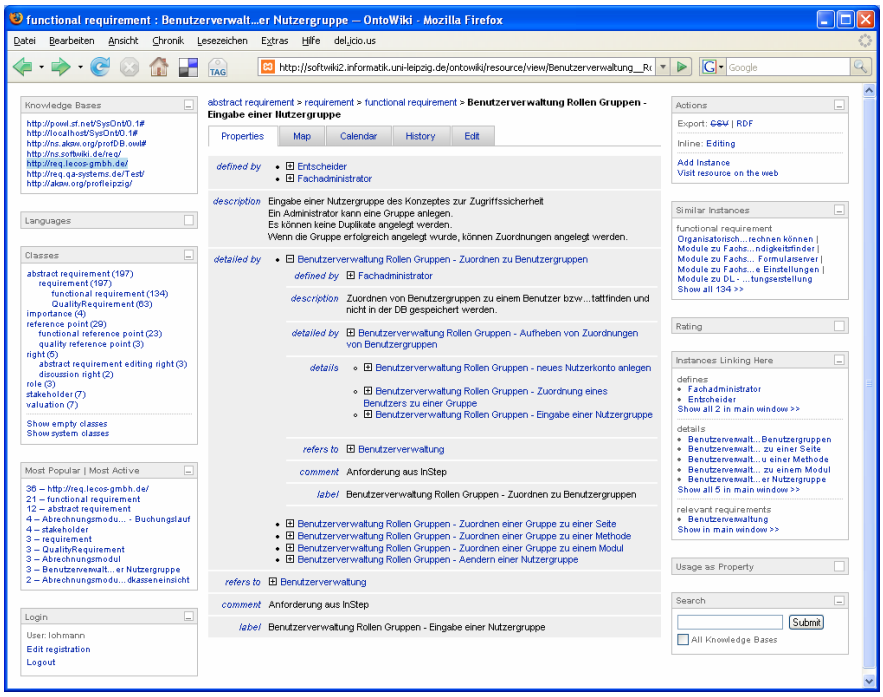


Figure 5: Functional requirement instance related to user roles and access control of the e-government application

In order to use OntoWiki for requirements elicitation we pre-loaded the SWORE ontology into this platform. The existing requirements have been imported directly from the prior used software management tool. Stakeholders are now enabled to create and interlink requirements and scenarios. Furthermore, they can vote, discuss, and comment individual pieces of information. Figure 5 shows a functional requirement instance related to user roles and access control of the e-government application.

The initial project ontology has been created in a combined method: (1) by an analysis of the site map of the existing web application, (2) by the use of concept extraction methods based on statistical text mining procedures [Bö02].

[5] OntoWiki is open source software and available at http://ontowiki.net.

Figure 6 shows a tag cloud visualization [KL07] that has been generated from the reference points of the use case. A reference point's font size represents its usage popularity. Though this visualization type has limited expression capabilities, it provides a good impression of major concepts with respect to the project and the application that is to be developed. It furthermore facilitates the exploration and navigation of the requirements: simply by clicking on a tag (reference point) in the cloud, a stakeholder gets a list of all associated requirements.

Figure 6: Tag cloud visualization of the reference points

6 Conclusion and Future Work

We consider a combined approach based on Semantic Web and Social Software concepts as beneficial for requirements engineering, since a crucial part consists in the collaborative development of a common terminology and shared understanding between different stakeholders. The presented approach tries to facilitate stakeholder participation and to foster the development of a conceptual basis. Similar to the notion of Wikis and Social Tagging, stakeholders are enabled to explore and expand the project ontology from different angles leading to new insights and a better understanding regarding the planned software application. Our current activities include further investigation of the e-government use case and the preparation of two additional use cases in e-commerce and geographic information system (GIS) domains.

Literaturverzeichnis

[ADR06] Auer, S.; Dietzold, S.; Riechert, T.: OntoWiki – A Tool for Social, Semantic Collaboration. In: Proceedings of the 5th International Semantic Web Conference, Springer, 2006; pp. 736–749.

[AL07] Auer, S.; Lehmann, J.: What have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. In: Proceedings of the European Semantic Web Conference, ESWC2007, Springer, 2007; pp. 503-517

[Bec00] Kent Beck. Extreme Programming Explained – Embrace Change. Addision Wesley, 2000.

[Bö02] Böhm, K.; Heyer, G.; Quasthoff, U.; Wolff, C.: Topic Map Generation Using Text Mining. In: Journal of Universal Computer Science, 8(6), 2002; pp. 623-643

[BT95] Boland J.R., Tenkasi R.V.: Perspective Making and Perspective Taking in Communities of Knowing. In: Organization Science, 6(4), 1995; pp. 350-372

[KL07] Kaser, O.; Lemire, D.: TagCloud Drawing: Algorithms for Cloud Visualization. In: Proceedings of WWW 2007 Workshop on Tagging and Metadata for Social Information Organization, 2007.

[JS06] John, A.; Seligmann, D.: Collaborative tagging and expertise in the enterprise. In: Proceedings of WWW 2006 Workshop on Collaborative Web Tagging, 2006.

[Lam01] Van Lamsweerde, A.: Goal-Oriented Requirements Engineering: A Guided Tour. In: 5th IEEE International Symposium on Requirements Engineering (RE'01), IEEE Computer Society Press, 2001; pp. 249-263

[Ma06] Marlow, C.; Naaman, M.; Boyd, D.; Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. In: Proceedings of the 17th Conference on Hypertext and Hypermedia, 2006; pp. 31-40

[HPW98] Haumer, P.; Pohl, K.; Weidenhaupt, K.: Requirements Elicitation and Validation with Real World Scenes. IEEE Transactions on Software Engineering, 24(12), 1998; pp. 1036-1054.

[Poh07] Pohl, K.: Requirements Engineering – Grundlagen, Prinzipien, Techniken. Dpunkt, 2007.

[Roy87] Royce, W.W.: Managing the Development of Large Software Systems: Concepts and Techniques. In: Proceedings of the 9th International Conference on Software Engineering, 1987; pp. 328-338.

[Sa06] Sack, H.: Kollaborative Indexierung und die Emergenz neuer sozialer Netzwerke. In: Workshop Social Software in der Wertschöpfungskette, 2006

Discovering Unknown Connections – the DBpedia Relationship Finder

Jens Lehmann[1]
lehmann@informatik.uni-leipzig.de

Jörg Schüppel[1]
joergschueppel@web.de

Sören Auer[1,2]
auer@seas.upenn.edu

[1]Universität Leipzig
Department of Computer Science
Johannisgasse 26
D-04103 Leipzig, Germany

[2]University of Pennsylvania
Department of Computer
and Information Science
Philadelphia, PA 19104, USA

Abstract: The Relationship Finder is a tool for exploring connections between objects in a Semantic Web knowledge base. It offers a new way to get insights about elements in an ontology, in particular for large amounts of instance data. For this reason, we applied the idea to the DBpedia data set, which contains an enormous amount of knowledge extracted from Wikipedia. We describe the workings of the Relationship Finder algorithm and present some interesting statistical discoveries about DBpedia and Wikipedia.

1 Introduction

Technologies based on Semantic Web standards are applied to various areas inside and outside the World Wide Web. A fundamental task is the creation and extension of ontologies, e.g. using the OWL[1] ontology language. In this work, we present a new user interface allowing to visualise connections in ontologies with large amounts of instance data.

The goal of the DBpedia Relationship Finder[2] is to provide a user interface to explore the huge DBpedia data set[ABK+07] by providing a means to find connections between different objects. The background knowledge consists of all the facts, which have been extracted from Wikipedia, in particular the information extracted from infoboxes (see [AL07] for details). The resulting web application allows the user to enter two objects, which are described by articles in the English Wikipedia, and computes connections between them. The application makes heavy use of Web 2.0 concepts like AJAX. The connections are obtained by querying the RDF data of the underlying triple store. Therefore, the methods we propose and the user interface we develop can be used for arbitrary triple stores, as detailed in Section 5, but we will focus on DBpedia within this article. As a by-product of creating the Relationship Finder, we analysed the DBpedia RDF graph. We will present some interesting insights we gained.

[1]http://www.w3.org/2004/OWL
[2]available at http://wikipedia.aksw.org/relfinder/

Overall, the paper makes the following contributions:

- development of a new DBpedia user interface using Semantic Web and Web 2.0 techniques,
- statistical analysis of DBpedia and Wikipedia data,
- a new general RDF browsing interface.

The article is structured as follows: In Section 2 we give a brief overview of the DBpedia project. We then proceed to showing how we processed the obtained information in Section 3. The results of the preprocessing are used as input for the Relationship Finder and, furthermore, allow to derive some interesting statistics about DBpedia and therefore also Wikipedia. Section 4 describes the Relationship Finder algorithm and user interface. In Section 5 we give a different view of the DBpedia Relationship Finder as a general means to access the contents of RDF triple stores. We describe related work and conclude in Section 6.

2 The DBpedia Project

The DBpedia project[ABK+07] is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia and to link other datasets on the Web to Wikipedia data. The core of DBpedia is a method to extract RDF triples from the infobox templates used within Wikipedia articles (details can be found in [AL07]). Since Wikipedia authors developed templates which provide predefined information structures for a variety of domains the infobox dataset contains data for and relationships between entities from a multiplicity of knowledge domains. These include cities (4,872), music albums (35,190), people (19,834), books as well as information about special interest domains such as computer games (365), planes (527) or amphibians (736).

The sheer amount of multi-domain data of the infobox extraction dataset and the inability of existing tools to handle this amount of data inspired the development of the Relationship Finder and builds its basis. Besides information extracted from infoboxes, the DBpedia project also provides datasets containing various other information and metadata about Wikipedia articles, e.g. article abstracts, information about labels (in different languages), images and links related to articles and categories. All these datasets are provided for download as RDF dumps. They are accessible as linked data[BCH07] and available for querying in form of an SPARQL endpoint.

The DBpedia project also aims to be a hub for user interfaces visualizing DBpedia data for easy access and browsing by human users. The project comprises a query builder, a combined full-text and facet-based search interface and is browsable with linked-data browsers such as Tabulator[3] or Disco[4]. The availability of the DBpedia data in various

[3] http://www.w3.org/2005/ajar/tab

[4] http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/disco/

forms already stimulated many people to create mashups or specialized user interfaces. Despite its short time of existence, the DBpedia project already evolved into a crystallisation point for knowledge bases on the Web. The DBpedia datasets are interlinked with ontologies and knowledge bases such as Wordnet, Musicbrainz and Revyu.

3 Decomposition Algorithm and Statistical Discoveries

This section describes how we pre-processed the DBpedia RDF data to apply the Relationship Finder on it. Note, that the Relationship Finder can work even without this preprocessing step. However, some of its features will not be available in this case. The source code of the algorithms presented here and in the following sections are available within the DBpedia sourceforge project[5].

Algorithm 1: RDF Graph Decomposition.

Input: an RDF statements table (a set of triples)
Output: objects separated in components stored in a component table

```
1  create necessary database tables;
2  filter triples in the statements table and copy them in a table T;
3  initialise an empty queue Q;
4  clusterId = 0;
5  while T is not empty do
6      pick first object O from T and add it at the end of Q;
7      write O to component table;
8      while Q is not empty do
9          find all objects obj, which are object or subject of a triple in T, which contains O
           as subject or object;
10         forall O' ∈ obj do
11             if O' ∉ Q then
12                 add O' at the end of Q;
13                 add O' to component table;
14             delete triples in T containing O';
15         set O to first object in Q;
16     increment clusterId;
```

The goal of the pre-processing can be described as follows: We treat the extracted DBpedia infobox graph as an undirected graph and want to find its components, i.e. its maximal connected subgraphs. (Two objects are in the same component if and only if there exists a path between them.) Given two objects, this allows us to decide whether they are connected in the underlying RDF graph. If they are not connected, the Relationship Finder can terminate immediately. If they are connected, we want to be able to find a path between the

[5] http://dbpedia.svn.sourceforge.net/viewvc/dbpedia/relfinder/

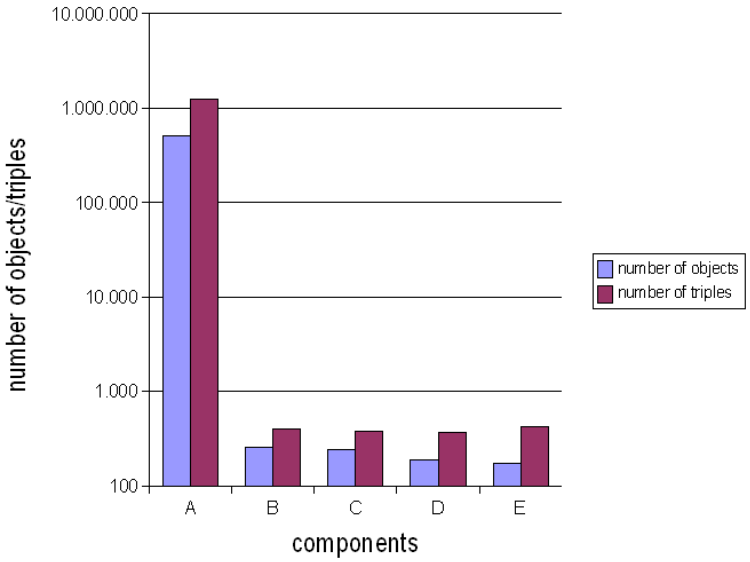


Figure 1: Number of triples and objects in the five largest components (ordered by number of objects).

two objects. While finding the shortest path is the computational most expensive part of the Relationship Finder algorithm, the pre-processing allows us to derive a (not necessarily shortest) path between two objects.

Algorithm 1 shows how we decompositioned the RDF graph. It works on an RDF statements table. Before applying the algorithm, we filter all triples, which we want to use, e.g. those not containing literal values, and copy them into a separate table. This filter can be configured to include or ignore certain types of triples if desired. The filtered DBpedia infobox data set still contains 1.5 million triples. We start from an arbitrary object and use a breadth first strategy to find all connected objects, i.e. all other objects within the component. The decomposition results are stored in database tables. For each object, we store its component id (the id of the component it belongs to), the minimum distance from our starting object within the component, and the object and property linking to the next object on the path to the starting object within the component. Please note, that the algorithm itself is not novel, but a straightforward application of existing techniques to RDF triple stores.

We determined that on average each DBpedia object has 5.67 outgoing connections, i.e. starting from an arbitrary object 5.67 other objects are directly connected. Considering outgoing connections of length two, 18 objects can be reached.

When running the cluster algorithm, we also generated some statistical information about the DBpedia components. Figure 1 shows the five largest components we obtained. Note the logarithmic scale on the y axis. We can see that almost all of the objects and triples are in the largest cluster. It accounts for 91% of all objects and 96% of all triples in the

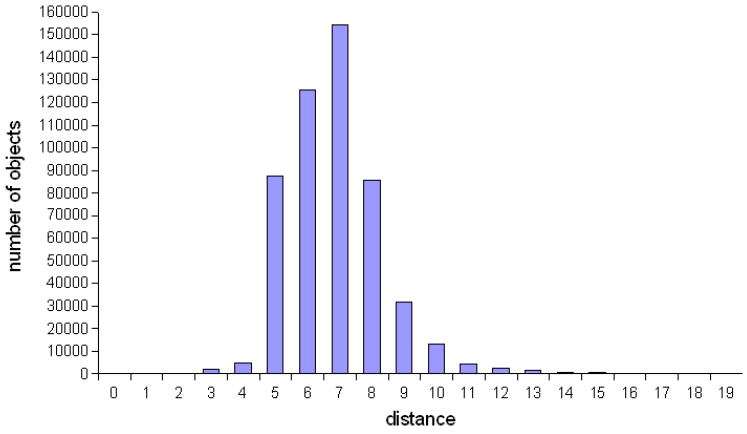


Figure 2: Number of objects with regard to the distance from a origin object.

DBpedia Infobox data set. This indicates that the DBpedia data set is densely connected, because starting from an arbitrary object in DBpedia (and therefore Wikipedia) this means you can reach almost any other object through properties in the DBpedia Infobox data set. This is one reason why it is interesting to construct a Relationship Finder, which allows to uncover these often interesting and surprising connections.

Figure 2 is another indicator of the density of the DBpedia RDF graph. As shown in Algorithm 1, we compute the components by starting with an arbitrary object. The figure shows the distance of any object in the main cluster from this starting object. Almost all of the objects have a distance between 5 and 9 from the starting object and are, thus, within a short distance from the starting object. The figure has to be interpreted cautiously, because it depends on the (randomly selected) starting object. A more comprehensive analysis is subject to further work.

4 The Relationship Finder

This section describes the actual Relationship Finder web application based on the decomposition introduced before. We will first describe the user interface and then explain the underlying algorithm.

User interface. The Relationship Finder user interface is very intuitive. Initially, it contains a simple form to enter two objects, as well as a small number of options, and a list of previously saved queries. For entering objects, the user can utilize the autocompletion feature (see Figure 3), which is implemented using the freely available Scriptacolous JavaScript library[6]. While typing, the user is offered suggestions for the object he wants

[6]http://script.aculo.us/

to enter. The corresponding database queries are performed in the background and loaded into the displayed Web page using AJAX technology. After submitting the query by clicking the ”find relation” button, the Relationship Finder algorithm starts. First, the user is informed whether a connection between the objects exists. If such a connection exists, the user can, furthermore, preview a connection between the objects (see Figure 4). The details of this procedure are explained later.

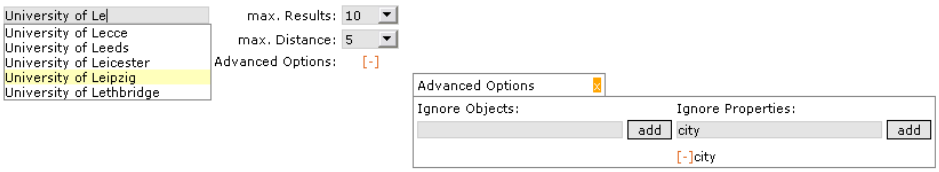


Figure 3: Autocompletion feature.

The preview connection does not have the guarantee to be the shortest available connection. For this reason, the Relationship Finder tries to find shorter solutions. When it has finished the computation, a configurable number of different connections is presented. Each connection is shown as a path where the leftmost part is the first entered object and the rightmost part is the second entered object. In between are the objects and properties, which connect these. Note, that the property arrows go in both directions, because we treat the underlying RDF graph as undirected.

Every object contains two additional buttons: The first one opens a box, which displays the available knowledge about this object. This information is retrieved from the statements database table using AJAX. The box parses information into user friendly formats, e.g. connections to other objects are shown as links, connection to images are directly displayed as image, and lists are recognised and displayed as such. All objects are transformed to links to the corresponding Wikipedia articles. The second button, which is associated with each object (and also each property in this case) is an ignore button depicted by a red cross. This allows to add objects and properties to an ignore list, i.e. the user states that in the next query he wants to ignore all connections containing these objects or properties. This list can also be edited by hand, again using autocompletion as a useful feature. Figure 5 shows a screenshot where both additional buttons are used.

After a query has been executed, the user can save it to make it available for other users. It is then displayed in the list of previously saved queries. This list can be ordered by popularity and query creation time. The results of these saved queries are cached, such that no significant server load is caused by executing these queries several time.

Technical Implementation. We assume that the components have been computed as described in Section 3. Another pre-processing step is to generate an undirected version of the statements table. This means that for each S-P-O triple, another O-P-S triple is written. This is done, because we consider the underlying RDF graph as undirected. The main reason why we are performing this as a pre-processing step (instead of the core algorithm) is efficiency.

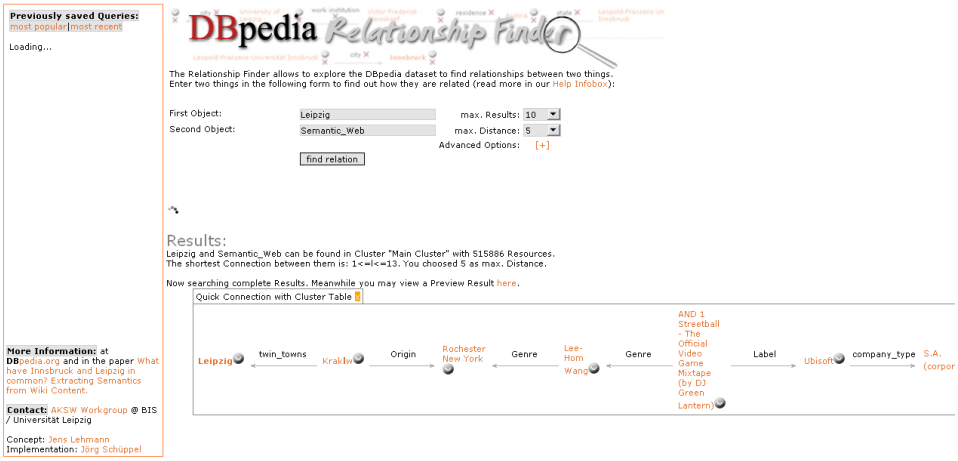


Figure 4: Precomputed connection between two objects.

Algorithm 2 shows the base structure of the Relationship Finder algorithm. Some parts will be explained in more detail in the next paragraphs.

Line 2 states that a minimum and maximum distance between two objects O_1 and O_2 according to the components table are computed. This is done as follows: Let O_S be the starting object in the component of O_1 and O_2. From the components table we can obtain the two paths from O_1 to O_S and O_2 to O_S, respectively. The minimum distance *min* is then:

$$min = |distance(O_1, O_S) - distance(O_2, O_S)|$$

This follows from the fact that we used breadth first search in Algorithm 1, i.e. we know that the computed distances between an object and the starting object within a component are minimal. Say $distance(O_1, O_S) < distance(O_2, O_S)$ (without loss of generality), then the existence of a path with length smaller than *min* between O_1 and O_2 would imply that $distance(O_2, O_S)$ is not minimal, which is a contradiction.

Similarly, the maximum length is the sum of the distances. However, in this case we can give a better estimate. We can look for objects, which the paths from O_S to O_1 and O_S to O_2 have in common. If O_C is such an element, then $O_1 - \ldots - O_C - \ldots - O_2$ is a possible path from O_1 to O_2. We pick the common element, which minimises the length of such a path (due to O_S there is always a common element). The path we obtain is the one shown in the preview of the DBpedia Relationship Finder and its length is an upper bound of the length of the shortest path between O_1 and O_2.

The next interesting part of the algorithm is line 2, which generates the SQL database query to find the connections. The generated query contains JOIN operations corresponding to the current distance we are interested in. The underlying database systems usually optimise these operations, such that the JOINS are executed in an efficient order. However,

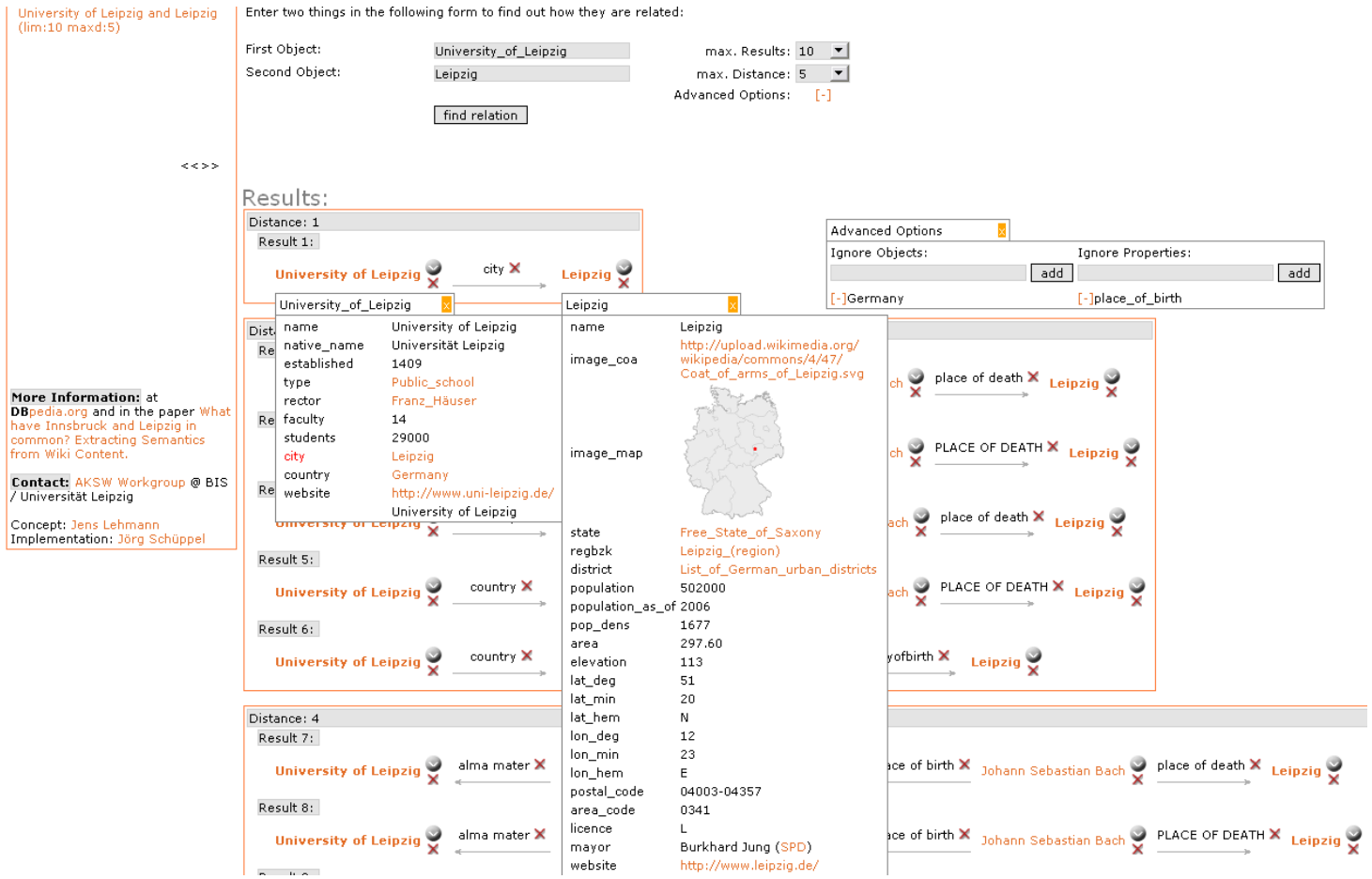


Figure 5: Boxes with additional information and ignore list options in the Relationship Finder.

Algorithm 2: Workings of the DBpedia Relationship Finder.

Input: first object O_1, second object O_2, maximum distance d_{max}, maximum number of results n, ignore list of objects and predicates

```
if query has been saved then
    load result from cache;
else
    if O_1 and O_2 are in the same component then
        compute minimum distance min and maximum distance max according to components table;
        compute preview connection and display it;
        set d = min;
        set m = 0;
        while d < d_max and m < n do
            formulate SQL query for obtaining at most (n − m) connections between O_1 and O_2 of length d without objects and properties in the ignore list;
            if connections exist then
                display connections;
                m = m + number of found connections;
            increment d;
            if d = d_max then
                Output: no connections within the specified maximum distance exist
    else
        Output: no connection exists, objects in different components
```

depending on the query, these operations can still be very expensive for high distances, which is why we limit the distance between the two objects to 10. The SQL query is extended by constructs, which forbid double occurrences of objects and properties within a connection. Furthermore, the ignore lists are also taken into account here, i.e. we extend the query to disallow any of the objects and properties to be ignored in the connection.

5 The Relationship Finder as RDF Userinterface

This section gives some remarks about the use of the Relationship Finder for general RDF knowledge bases. As noted before, DBpedia is just one interesting application of the Relationship Finder. However, except for some DBpedia specific features (e.g. links to Wikipedia corresponding articles) it is not restricted to DBpedia. This is a brief overview of existing techniques for visualising ontology instance data:

- graphs: tools for navigating along RDF graphs
- tables: data organised in tabular form

- triples: data shown as basic triples
- timetables: usage of time oriented presentations, e.g. in personal organisers
- maps: usage of place oriented presentations, e.g. showing data in maps
- mashups: data collected from various sources and displayed together

The Relationship Finder extends this list by displaying a set of paths between two objects in an RDF graph of interest. A path here can be seen as a selection of interesting triples. The Relationship Finder is a general purpose user interface (such as graph or triple visualizations). It is especially suited for knowledge bases, which do not allow other visualization forms (such as graph or triple visualizations) due to their sheer amount of data.

6 Related Work, Conclusions, and Further Work

Related Work. We will first describe work related to the DBpedia project and afterwards work, which describes interfaces to RDF knowledge bases.

Apart from the DBpedia project, there have been other attempts to extract information from Wikipedia and make it available for further use. YAGO [SKW07] is an effort, which extracts 14 relations from the Wikipedia category system, Wikipedia redirects, and other sources of information within Wikipedia. Freebase[7] is a project by MetaWeb[8], which has the goal to build up a huge database of editable information. They used Wikipedia to reach an initial critical mass of information. Semantic MediaWiki [KVV05, VKV+06] is an extension of the MediaWiki software, which is the Wiki software underlying Wikipedia. It allows to add structured data based on RDF to Wikis, which enables information reuse as well as enhanced search and browse facilities.

Techniques for discovering relationships between objects within knowledge bases were for example also developed in the course of the SemDis project[9]. In [AMHWA+05] for instance, a flexible ranking approach is presented which can be used to distinguish more interesting and relevant relationships from less important ones. In [AMNR+06], similar techniques were applied to address the problem of conflict of interest detection by analysing social networks.

Conclusions. We presented a novel RDF user interface, which is especially applicable to ontologies with large amounts of instance data. As one example for such an ontology, we used the DBpedia infobox data set. We implemented our approach and made it available online. We incorporated the feedback and feature requests we obtained in this application. The Relationship Finder uses a combination of existing algorithms in the background, AJAX technologies for providing a responsive and user-friendly interface,

[7] http://www.freebase.com
[8] www.metaweb.com/
[9] http://lsdis.cs.uga.edu/projects/semdis/

and numerous features like saving and caching queries, ignoring objects, and presenting additional information about objects.

Future Work. Possible lines of future work are to extend the Relationship Finder from DBpedia infoboxes to other parts of the DBpedia data set, to apply the Relationship Finder to other knowledge bases, and to improve our analysis of DBpedia/Wikipedia data.

References

[ABK+07] Sören Auer, Chris Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, 2007. To appear.

[AL07] Sören Auer and Jens Lehmann. What Have Innsbruck and Leipzig in Common? Extracting Semantics fromWiki Content. In *Proceedings of the 4th European Semantic Web Conference (ESWC)*, pages 503–517, 2007.

[AMHWA+05] Boanerges Aleman-Meza, Christian Halaschek-Wiener, Ismailcem Budak Arpinar, Cartic Ramakrishnan, and Amit P. Sheth. Ranking Complex Relationships on the Semantic Web. volume 9, pages 37–44, 2005.

[AMNR+06] Boanerges Aleman-Meza, Meenakshi Nagarajan, Cartic Ramakrishnan, Li Ding, Pranam Kolari, Amit P. Sheth, Ismailcem Budak Arpinar, Anupam Joshi, and Tim Finin. Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *WWW*, pages 407–416. ACM, 2006.

[BCH07] Christian Bizer, Richard Cyganiak, and Tom Heath. *How to publish Linked Data on the Web*. http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/, 2007.

[KVV05] Markus Krötzsch, Denny Vrandecic, and Max Völkel. Wikipedia and the Semantic Web - The Missing Links. In Jakob Voss and Andrew Lih, editors, *Proceedings of Wikimania 2005, Frankfurt, Germany*, 2005.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A Core of Semantic Knowledge - Unifying WordNet and Wikipedia. In *16th International World Wide Web Conference (WWW 2007)*, Banff, Canada, 2007.

[VKV+06] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic Wikipedia. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th international conference on World Wide Web, WWW 2006*, pages 585–594. ACM, 2006.

Semantisch unterstütztes Requirements Engineering

Thomas Riechert[1], Kim Lauenroth[2], Jens Lehmann[1]

[1]Abteilung Betriebliche Informationssysteme
Institut für Informatik, Universität Leipzig
{riechert|lehmann}@informatik.uni-leipzig.de

Postfach 100920
04009 Leipzig

[2]Software Systems Engineering
Institut für Informatik und Wirtschaftsinformatik, Universität Duisburg-Essen
kim.lauenroth@sse.uni-due.de

Schützenbahn 70
45117 Essen

Abstract: Requirements Engineering ist eine erfolgsentscheidende Phase von Software-Entwicklungsprojekten, welche sich besonders dadurch auszeichnen, dass viele verschiedene Stakeholder gemeinsam Ziele, Szenarien und Anforderungen für das geplante System erheben. Neben der Entwicklung eines Vorgehens für ein semantisch unterstütztes Requirements Engineering, wurde eine Ontologie zur Abbildung anforderungsrelevanter Information entwickelt. Diese wird zusammen mit einem Wissensmodellierungs-Werkzeug, anhand eines realen Anwendungsfalls aus dem Bereich E-Government, beschrieben.

1 Ausgangssituation

Das Requirements Engineering (RE) stellt eine entscheidende Phase innerhalb des Software-Entwicklungsprozesses dar. Die Technologie des RE werden aber auch in einem allgemeineren Kontext im Projekt-Management eingesetzt. Die Bedeutung des RE für den Erfolg eines Projektes wurde durch mehrere Studien (u.a. [Hall et. Al 2002]) belegt.

Ein zentrales Ziel des Requirements Engineering ist die Entwicklung eines gemeinsamen Verständnisses über die Ziele, Szenarien und Anforderungen an das geplante System. In diesem Zusammenhang benötigen die verschiedenen im Entwicklungsprozess beteiligten Stakeholder eine gemeinsame Terminologie, mit der die Stakeholder sich ohne Missverständnisse verständigen können.

Innerhalb des Semantic Web existieren Standards für die Entwicklung und Anwendung von Terminologien auf verschiedene Domänen. Im Kontext des Semantic Web werden Taxonomien und Ontologien z.B. durch RDF[1], RDF-Schema[2] und OWL[3] ausgedrückt. Software- und Entwicklungsprojekte haben eine zunehmende Anzahl räumlich verteilter Stakeholder. Die Wissensrepräsentations-Standards bilden eine solide Basis für die verteilte Gewinnung, die Repräsentation, die Strukturierung und das Management von anforderungsrelevanten Informationen. Die semantische Repräsentation von anforderungsrelevanten Informationen kann ein Kristallisationspunkt für die Integration verschiedener Projektentwicklungswerkzeuge (wie Projektmanagement-Software und CASE-Werkzeuge) darstellen.

In diesem Artikel stellen wir einen Ansatz für semantisches RE dar. Nachdem grundsätzliche RE-Konzepte in Kapitel 2 vorgestellt werden, stellen wir ein Ontologieschema für das RE vor (Kapitel 3). In Kapitel 4 berichten wir über ein Werkzeug für semantisch basiertes RE und seine Anwendung in einem realen Anwendungsfall in der Domäne E-Government. Zukünftige Arbeiten werden in Kapitel 5 beschrieben.

2 Requirements Engineering

Requirements Engineering ist die Phase von Software- und Projektentwicklungs-Prozessen, in der die Anforderungen für ein geplantes System erhoben werden. Das Requirements Engineering ist ein kooperativer und iterativer Prozess, der versucht, die folgenden drei Ziele zu erreichen (vgl. [Pohl 1996], [Pohl 2007]):

Gewinnung aller relevanten Anforderungen: Zu Beginn des RE sind die Anforderungen der Stakeholder an ein geplantes System nicht unbedingt umfassend bekannt oder verstanden. Im RE werden daher Anforderungen von allen relevanten Stakeholdern ermittelt oder gemeinsam mit den Stakeholdern entwickelt.

[1] http://www.w3.org/RDF/
[2] http://www.w3.org/TR/rdf-schema/
[3] http://www.w3.org/2004/OWL/

Etablierung einer ausreichenden Übereinstimmung über die Anforderungen: Die Übereinstimmung von Anforderungen ist eine Voraussetzung für die Akzeptanz eines geplanten Systems. Konflikte unter den Stakeholdern gefährden die Übereinstimmung unter den Stakeholder. Folglich müssen im Rahmen des RE-Prozesses bestehende Konflikte identifiziert und z.B. durch Diskussionen, Abstimmungen oder Entscheidungen aufgelöst werden.

Projektkonforme Dokumentation der Anforderungen: Unterschiedliche Arten von Systemen erfordern unterschiedliche Arten der Anforderungsspezifikation. Zum Beispiel benötigen sicherheitsrelevante Systeme, wie ein Flugzeugsteuerungssystem, eine strenge und formale Definition von Anforderungen, um das Einhalten von sicherheitsrelevante Eigenschaften des System zu verifizieren. Infolgedessen ist es wesentlich im Requirements Engineering, die Anforderungen entsprechend den spezifischen Erfordernissen des geplanten Systems zu dokumentieren.

Ein genereller Ansatz für das Erreichen dieser Ziele, ist der Nutzung von Zielen und Szenarien im RE. Ziele beschreiben die Absichten der Stakeholder für das geplante System (cf. [Pohl et al. 1998]). Ziele unterstützen zum Beispiel die Konfliktlösung, da es einfacher ist, eine Vereinbarung über ein abstraktes Ziel als über eine detaillierte Anforderung zu erreichen. Szenarien beschreiben exemplarische Nutzungsabläufe des geplantes Systems, die zur Erfüllung oder Nicht-Erfüllung von Zielen führen (cf. [Pohl et al. 1998]). Szenarien erleichtern zum Beispiel die Gewinnung von Anforderungen indem Stakeholder die Nutzung des Systems an konkreten Beispielen beschreiben.

3 Requirements Engineering Ontologie

Um den RE-Prozess zu sematisch zu unterstützen, haben wir die semantische RE Ontologie SWORE – SoftWiki[4] Ontology für RE entwickelt. SWORE liefert eine semantische Struktur, die anforderungsrelevante Informationen aufnimmt und eine Verlinkung zu domain- und anwendungsspezifischem Vokabular herstellt.

[4] http://www.softwiki.de

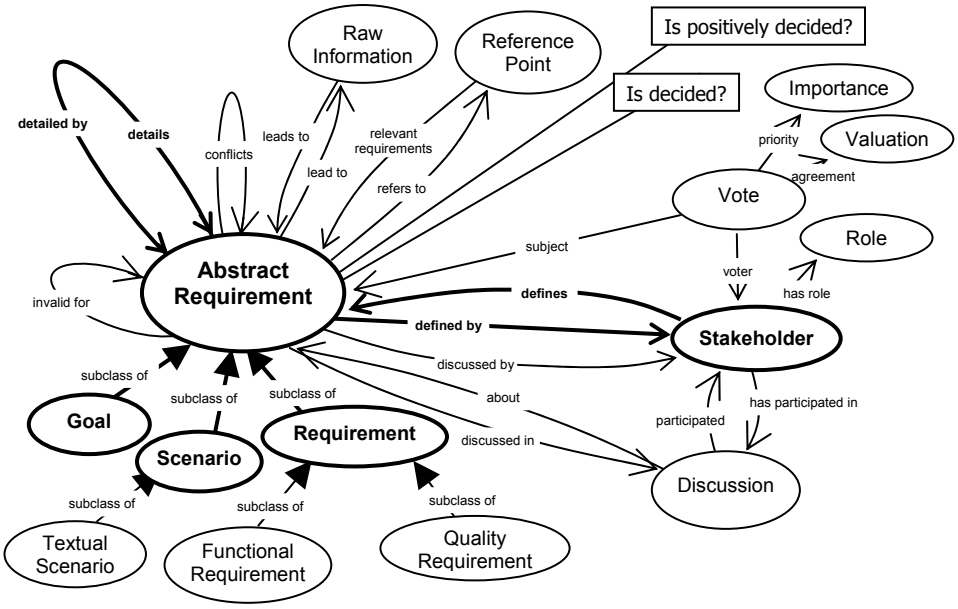


Abbildung 1: Visualisierung des Kerns der RE Ontologie

Abbildung 1 zeigt den Kern der SWORE Ontologie, welche in Anlehnung an etablierte Informationsmodelle des Requirements Engineering [Pohl et al. 1998, Pohl 1996, Pohl 2007, van Lamsweerde 2001] entwickelt wurde. Zentral für unseren Ansatz sind die Klassen Stakeholder und Abstract Requirement sowie die Properties *details* und *defines*. Abstrakte Requirements haben die die abgeleiteten Klassen Goal, Scenario und Requirement, von dem jedes von den Stakeholdern definiert werden und durch andere abstrakte Requirements detailliert werden kann. Dies ermöglicht die Spezifikation von Abstract Requirements unterschiedlicher Granularität. Wir heben die gemeinschaftlichen Aspekte des RE hervor, indem wir Diskussionen unter den Stakeholdern integrieren und Abstimmungen (mit den Klassen Agreement und Importance) im Modell aufgenommen haben. Im RE Prozess ist die Dokumentation häufig für zukünftige Entscheidungen relevant. Um Anforderungen mit bestehenden Dokumenten oder Resourcen zu verbinden enthält SWORE die Klassen Raw Informationen und Reference Point zusammen mit den verwendeten Properties. SWORE steht als Download unter http://softwiki.de/SWORE zur Verfügung.

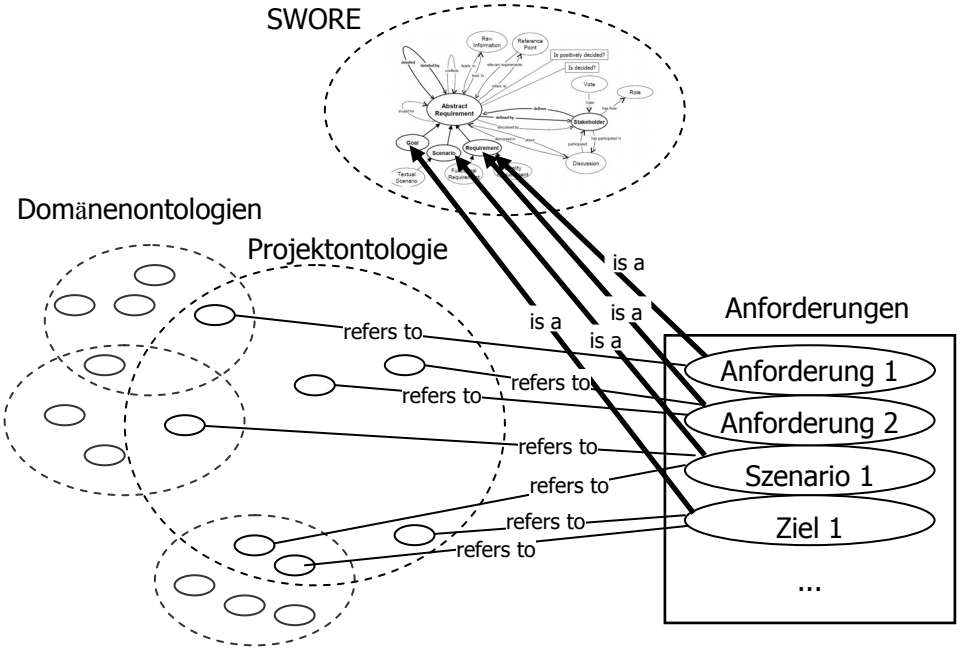


Abbildung 2: Anwendung von SWORE und die Verknüpfung mit der Projekt-Ontologie

Der Einsatz der SWORE wird in der Abbildung 2 dargestellt. Eine RE-Wissensbasis enthält Instanzen der SWORE Konzepte. Die Innovation des Verwendens der semantischen Darstellungen im Vergleich mit traditionellen RE-Werkzeugen ist die Möglichkeit, vorhandene Informationen und Domain-Ontologien wieder zu verwenden und einzubeziehen. In absehbarer Zeit werden immer mehr Domain Ontologien zur Verfügung stehen. Diese sind für den RE-Prozess sehr wichtig, da sie das Verständnis zwischen Stakeholdern erleichtern. Das ist ein wesentlicher Bestandteil des geplanten Einsatzes der SWORE.

4 E-Government Anwendungsfall

Eine Anwendung des vorgestellten semantisch basiertem RE wird in einem E-Government Anwendungsfall demonstriert. Deutsche Einwohner müssen ihren gegenwärtigen Wohnort bei der der lokalen Stadtverwaltung registrieren. Aus historischen Gründen entstand eine heterogene Systeminfrastruktur, um den automatisierten Datenaustausch zwischen unterschiedlichen Stadtverwaltungen zu erleichtern und Selbstbedienungssysteme für Bürgeramtsdienstleistungen zur Verfügung zu stellen, finden Bemühungen diese einzelnen Anwendungen auf einer lokalen, regionalen und Bundesebene zu homogenisieren und besser zu integrieren, statt.

Wir nutzen den vorliegenden Anwendungsfall um die Funktionsweise der Methode mit realen RE- Daten zu demonstrieren. Das vorliegende Projekt ist bereits implementiert und wird derzeit für den Einsatz in den Bürgerämtern der Stadt Leipzig vorbereitet. Günstigerweise lagen bereits Datenquellen für den Import in eine Ontologie vor. Dabei konnten alle Anforderungen aus dem für die Entwicklung verwendeten Management Tool in-Step[5] übernommen werden. Vorteilhaft erwies sich dabei die Tatsache, dass diese bereits in verschiedenen Detailierungsgraden vorlagen. Stakeholder konnten aus den Dokumentationsdokumenten extrahiert werden. Eine Weitere Basis für Referenzpunkte bildet die Side-Map der Webapplikation.

Abbildung 3: Darstellung eines funktionalen Requirements im OntoWiki

Das Problem der Integration dieser unterschiedlichen Anwendungen ist jedoch nicht nur ein technisches, da unterschiedliche Verwaltungen unterschiedliche Prozesse und Strukturen etabliert haben. Folglich haben sie unterschiedliche Anforderungen im Rahmen der Integration und des Datenaustausches. Um den verschiedenen Stakeholdern ein kollaboratives Erheben von Anforderungen zu ermöglichen, erweitern wir die semantische Kollaborations-Plattform OntoWiki [Auer et al. 2006].

[5] http://www.microtool.de/instep/de/

OntoWiki ist eine semantische Webanwendung, die einer verteilten Benutzergemeinschaft die Entwicklung von Ontologien und das Sammeln von zugehörigen Daten ermöglicht. OntoWiki strebt ein möglichst einfaches Browsen in Wissenbasen und in den Benutzerbeiträgen an. OntoWiki ist Open Source und steht als Download unter http://ontowiki.net zur Verfügung.

Um OntoWiki für die die Erhebung von Anforderungen verwenden zu können, muss als Erstes die SWORE geladen werden. Danach sind die Stakeholder in der Lage, Anforderungen und Szenarien zu erstellen und zu verknüpfen. OntoWiki stellt auch Funktionen zur Abstimmung, Diskussion und Kommentierung einzelner Informationen zur Verfügung und kann daher den Abstimmungsprozess unterstützen. Abbildung 3 zeigt eine funktionale Anforderung „Benutzerverwaltung Rollen Gruppen – Eingabe einer Nutzergruppe“.

Der Einsatz von Ontowiki ermöglicht die schrittweise Entwicklung anforderungsrelevanter Informationen in einer Wissensbasis. Dem Wiki-Grundsatz von OntoWiki folgend, schreibt OntoWiki keinen speziellen Prozess vor. Stattdessen können Anforderungen verfeinert werden, kommentiert, besprochen und jederzeit verknüpft werden, bis die Wissensbasis einen stabilen Zustand erreicht, dem die Stakeholder zustimmen.

5 Zusammenfassung und Ausblick

Die Anwendung von Semantic Web Technologien ist für RE-Prozesse vorteilhaft, da der Aufbau einer gemeinsame Terminologie zwischen den Stakeholder ein erfolgsentscheidender Faktor.

Aktuell versuchen wir mit weiteren RE Stakeholder Communities Erfahrungen aufzubauen und dabei das generische Werkzeug Ontowiki für diesen Zweck zu erweitern. Insbesondere möchten wir Schnittstellen zu RE- und Management Werkzeugen, wie IRqA[6] und Doors[7] entwickeln. Der Informationsaustausch mit solchen traditionellen Werkzeugen ist, da wir hauptsächlich auf die frühen Stadien Anforderungserhebung mit vielen räumlich verteilten Stakeholdern fokusieren. Sobald eine Übereinstimmung innerhalb einer Stakeholder Community erreicht wird, können die gewonnenen Informationen innerhalb der traditionellen Werkzeuge verwendet werden, um den restlichen Software- oder Projektentwicklungs-Prozess zu unterstützen.

Dieser Artikel ist eine Übersetzung von [Riechert et al. 2007] ins Deutsche mit Erweiterungen im Detail.

[6] http://www.irqaonline.com
[7] http://www.telelogic.com/products/doors

Danksagung

Die Autoren danken Dirk Fritzsch (QA-Systems GmbH), welcher große praktische Erfahrungen als Konsultant für RE-Projekte in die Unterstützung und Beratung bei der Erstellung des Ontologie-Schemas, einbrachte, sowie Steffen Lohmann, Universität Duisburg-Essen, für sein Feedback.

Literaturverzeichnis

[Auer et al. 2006] Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki - A tool for social, semantic collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, volume 4273, pages 736-749. Springer, 2006.

[Riechert et al. 2006] Sören Auer, Thomas Riechert, Klaus-Peter Fähnrich. SoftWiki – Agiles Requirements-Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder. GeNeMe'06 - Gemeinschaft in neuen Medien, 29. Sep. 2006, Dresden, Germany.

[Riechert et al. 2007] Thomas Riechert, Kim Lauenroth, Jens Lehmann, Sören Auer. Towards Semantic Based Requirements Engineering. In: Proceedings of the 7th International Conference on Knowledge Management (I-Know), Graz, Austria, 2007. *To appear.*

[Pohl et al. 1998] Peter Haumer, Klaus Pohl, and Klaus Weidenhaupt. Requirements elicitation and validation with real world scenes. IEEE Transactions on Software Engineering, 24(12):1036–1054, December 1998.

[Pohl 1996] Klaus Pohl. Process-centered requirements engineering. Research Studies Press, Advanced Software Development, 1996.

[Pohl 2007] Klaus Pohl. Requirements Engineering - Grundlagen, Prinzipien, Techniken. Dpunkt Verlag, 2007.

[Hall et al. 2002] A. Rainer T. Hall, S. Beecham. Requirements problems in twelve companies – an empirical analysis. In Proceedings of the 6th International Conference on Empirical Assessment and Evaluation in Software Engineering (EASE 2002). Keele University, 2002.

[van Lamsweerde 2001] Axel van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In Fifth IEEE International Symposium on Requirements Engineering (RE'01), pages 249-263. IEEE Computer Society Press, August 2001.

Ein Prozessmodell für ein agiles und wiki-basiertes Requirements Engineering mit Unterstützung durch Semantic-Web-Technologien[1]

Mariele Hagen[1], Berit Jungmann[2], Kim Lauenroth[3]

[1] PRO DV Software AG, Hauert 6, 44227 Dortmund
mariele.hagen@prodv.de
[2] T-Systems Multimedia Solutions GmbH, Riesaer Str. 5, 01129 Dresden
berit.jungmann@t-systems.com
[3] Software Systems Engineering, University of Duisburg-Essen, 45117 Essen
kim.lauenroth@sse.uni-due.de

Abstract: Requirements Engineering (RE) spielt in der Software Entwicklung eine entscheidende Rolle. Verschiedene Studien haben gezeigt, dass viele Projekte durch unzureichendes RE scheitern. Aus diesem Grund ist eine gezielte Anforderungsanalyse mit der Einbindung aller relevanten Stakeholder erfolgsentscheidend. Existierende klassische Tools unterstützen jedoch die kreative Phase des Requirements Engineering bisher nur ungenügend und setzen erst dann an, wenn bereits grobe Projektziele und Anforderungsstrukturen definiert wurden. Wiki-basierte Anwendungen scheinen dagegen für dieses Szenario eine geeignete Unterstützung darzustellen, insbesondere dann, wenn inhaltliche Abhängigkeiten zwischen Anforderungen zusätzlich durch Semantic Web-Technologien beschrieben werden. Der vorliegende Beitrag untersucht den Umgang mit großen und verteilten Stakeholdergruppen im RE und zeigt Verbesserungspotenziale auf. Im Rahmen des Projektes SoftWiki wurde das wiki-basierte Requirements Engineering als Lösungsansatz entwickelt, um Requirements Engineering mit großen und verteilten Stakeholdergruppen zu betreiben. Dieser Beitrag stellt das wiki-basierte Requirements Engineering vor und zeigt weiterführenden Forschungsbedarf auf.

1 Motivation

Softwareentwicklung ist in der Vergangenheit zu einem weltweiten Geschäft geworden. Zum einen wird Software weltweit verkauft, zum anderen wird Software weltweit entwickelt („The global connectivity provided by the Internet (…) drives both an organization's product and process strategies." [Boeh06], S. 22). Dies bedingt, dass eine ständig wachsende Anzahl von weltweit verteilten Stakeholdern im Requirements Engineering (RE) berücksichtigt und einbezogen werden muss. Darüber hinaus werden kurze Innovationszyklen und kurze Produkteinführungszeiten (time-to-market) zu einem

[1] Diese Arbeit wurde teilweise gefördert durch das BMBF-Projekt SoftWiki, Förderkennzeichen 01ISF02C|D|F.

entscheidenden Wettbewerbsfaktor, wodurch sich der verfügbare Zeitrahmen für die Entwicklung eines Systems (oder einer neuen Systemversion) und damit auch für das Requirements Engineering verringert.

Unternehmen beschäftigen sich zunehmend mit der Frage, inwiefern wiki-basierte Anwendungen für große und verteilte gemeinschaftliche Entwicklungsprozesse mit vielen beteiligten Stakeholdern Unterstützung leisten können. In diesem Zusammenhang wird auch der Begriff „Social Software“ angewandt, der den Aufbau sowie die Publikation und Verteilung von Informationen innerhalb sozialer Netzwerke fokussiert (weitere Ausführungen siehe [HiWi05]). Während Wiki-Anwendungen eine einfach zu organisierende Austauschplattform darstellen, sind semantische Verknüpfungen oftmals nicht möglich. Aus diesem Grund sind bereits erste semantische Wikis entstanden, die beide Trends verbinden (d.h. Social Software und Semantic Web, siehe erste Beispiele aus dem Wissensmanagement oder semantische Wikipedia-Ansätze [Sem07]).

Zur Identifikation der Probleme im Requirements Engineering mit großen und verteilten Stakeholdergruppen wurden vier Unternehmen bzgl. ihres Vorgehens im RE befragt. Es wurde u.a. festgestellt, dass die Gewinnung und Bewertung von Anforderungen mit großen und verteilten Stakeholdergruppen sehr aufwendig und kaum zu bewältigen ist. Auch kommerziell verfügbare Werkzeuge bieten hier nur wenig Unterstützung. Diese Werkzeuge (z.B. Doors, Irqa, RequisitePro) fokussieren die Dokumentation und das Management von Anforderungen. Sie leisten jedoch, so ein Ergebnis der Ist-Analyse, kaum Unterstützung für große und räumlich verteilte Stakeholdergruppen speziell in der frühen Phase des Requirements Engineering.

Dieser Beitrag widmet sich der Fragestellung, inwiefern ein wiki-basierter Requirements-Engineering-Prozess das Requirements Engineering mit großen und verteilten Stakeholdergruppen unterstützen kann. (s. auch [AuFa06, DeRR07]).

Der vorliegende Beitrag ist wie folgt strukturiert. Kapitel 2 beschreibt die Ergebnisse der Ist-Analyse unter den Industriepartnern. Kapitel 3 gibt einen Überblick über den Stand der Forschung und Kapitel 4 erläutert die Einbettung des wiki-basierten Requirements Engineering in den Software-Entwicklungsprozess. Kapitel 5 beschreibt das wiki-basierte Requirements Engineering und Kapitel 6 schließt diesen Beitrag mit einer Zusammenfassung und einem Ausblick auf weitere Forschungstätigkeiten.

2 Ist-Analyse zum Umgang mit großen und verteilten Stakeholdergruppen

Im Verbundprojekt „SoftWiki“ [Bmbf07, AuFa06] wurde eine Analyse zum Stand des Requirements Engineering bei den Industriepartnern durchgeführt. Hierbei stand insbesondere die Unterstützung großer, verteilter Stakeholdergruppen im Vordergrund. Es wurden Daten von 13 Projekten der vier im Projekt beteiligten Industriepartner erfasst und ausgewertet. Die befragten Projektpartner sind in verschiedenen Anwendungsbereichen (z. B. E-Commerce und Geoinformationssysteme) tätig.

Zur Verdeutlichung der Problematik des Requirements Engineering bei diesen Stakeholdergruppen werden zwei der analysierten Beispielprojekte betrachtet (Abschnitt 2.1). Anschließend werden die bei der Analyse identifizierten Probleme beschrieben (Abschnitt 2.2) und Verbesserungspotenziale aufgezeigt (Abschnitt 2.3).

2.1 Beispiele aus der Praxis

Bei der PRO DV Software AG wurde für das Bundesamt für Katastrophenhilfe und Bevölkerungsschutz (BBK) ein Projekt zur Entwicklung des deutschen Notfallvorsorgeinformationssystems (deNIS IIplus) durchgeführt. deNIS IIplus dient nicht nur zum Management von Großschadenslagen auf Bundesebene, sondern ebenso zur Erfassung und Übermittlung von Lagemeldungen auf der Ebene der unteren Katastrophenschutzbehörden [Deni07]. Dies bedeutet, dass mehrere größere Stakeholdergruppen, die zudem räumlich verteilt agieren, in das Requirements Engineering einbezogen werden mussten. Hierzu zählten u.a. das Bundesamt für Bevölkerungsschutz und Katastrophenhilfe (BBK) auf Bundesebene, das Land Hamburg auf Landesebene, das Gemeinsame Melde- und Lagezentrum (GMLZ), das Technische Hilfswerk (THW) und die Akademie für Krisenmanagement, Notfallplanung und Zivilschutz (AKNZ). Das Requirements Engineering wurde aus diesen Gründen räumlich und auch zeitlich verteilt in Form von Interviews oder Workshops mit großem Aufwand durchgeführt. Die Anforderungen wurden mithilfe des PRO DV-eigenen Werkzeugs ReqManager dokumentiert.

Bei der T-Systems Multimedia Solutions GmbH werden pro Jahr ca. 860 Kundenprojekte durchgeführt (davon ca. 250 Softwareprojekte z. B. im Bereich E-Commerce). Exemplarisch für eine Vielzahl von Projekten ist ein abgeschlossenes Shop-Projekt zu nennen, in dem das Requirements Engineering mit verteilten und großen Nutzergruppen eine besonders wichtige Rolle spielte. Herausforderungen ergaben sich insbesondere durch die Abstimmung der Anforderungen mit den heterogenen Fachabteilungen auf Seiten des Kunden, die über fünf verschiedene Standorte in Deutschland verteilt waren. Neben der Ermittlung von Anforderungen stellte die Erarbeitung eines gemeinsamen Glossars eine sowohl erfolgsentscheidende als auch aufgrund der unterschiedlichen fachlichen Sichtweisen schwierige Aufgabe dar. Für das Anforderungsmanagement wurde hierbei die Software DOORS eingesetzt. Eine besondere Aufgabenstellung ergab sich aus Sicht des Requirements Engineering in der frühen Phase der Gewinnung der Anforderungen, die bisher nur unzureichend durch existierende Tools unterstützt wird. Bei der Einbeziehung der insgesamt 63 Stakeholder in das Projekt ergab sich ein hoher Abstimmungs- und Dokumentationsbedarf, der nur bedingt mit existierenden Tools unterstützt werden kann.

2.2 Identifizierte Probleme im Requirements Engineering mit großen Stakeholdergruppen

Folgende Probleme wurden in den betrachteten Projekten beim Requirements Engineering mit großen, verteilten Stakeholdergruppen identifiziert (vgl. auch [LaHa07]):

P1. *Hoher zeitlicher Aufwand*: Es werden nicht nur große, sondern auch verteilte Stakeholdergruppen befragt. Dies bedeutet einen hohen zeitlichen Aufwand für An- und Abreisen, die Durchführung und die Terminfindung.

P2. *Hohes Aufkommen von Anforderungen*: Durch die vielen Stakeholder wird eine hohe Zahl von Anforderungen identifiziert. Dies erfordert einen hohen Aufwand bei der Dokumentation und dem Management der Anforderungen.

P3. *Große Ähnlichkeit von Anforderungen*: Obwohl es sich um verschiedene Stakeholdergruppen handelt, werden vielfach Duplikate erhoben oder ähnliche Anforderungen identifiziert. Die Identifikation von solchen Duplikaten und Varianten erfolgt manuell und ist sehr aufwändig.

P4. *Abstimmung unter den Stakeholdern*: Der Abstimmungsprozess über die Anforderungen gestaltete sich schwierig, da nie alle beteiligten Stakeholder gemeinsam über die Anforderungen diskutieren konnten, sondern die Abstimmung stets nur mit einer kleinen Gruppe von Stakeholdern erfolgte.

P5. *Hohe Anzahl von Feedback-Schleifen*: Die Konsolidierung der Anforderungen mit kleinen Gruppen bedingt viele Iterationen im Requirements Engineering, um alle Stakeholder über alle Änderungen an den Anforderungen zu informieren.

2.3 Verbesserungspotenzial im Requirements Engineering

Für die verteilte (virtuelle) Zusammenarbeit bedarf es Lösungen, die den kreativen Prozess und die Kommunikation im Requirements Engineering effektiv unterstützen. Bei der Betrachtung der Ist-Situation und in weiterführenden Interviews haben die Industriepartner u. a. folgende Verbesserungspotenziale genannt:

- Verstärkte und einfachere Einbeziehung aller Stakeholder in den Prozess (z.B. durch Einsatz von Wikis)

- Verbesserung der Transparenz durch Rückverfolgbarkeit von Änderungen (Traceability), d.h. Versionierung von Anforderungen

- Vermeidung von Medienbrüchen durch die Zusammenführung aller anforderungsrelevanten Informationen in eine Wissensbasis statt der Verwendung von E-Mails, Dokumenten und spezialisierten Tools

- Möglichkeit der Ordnung von Vorstellungen über das geplante System und finden eines gemeinsamen Vokabulars/einer gemeinsamen Domäne trotz räumlicher Distanz

- Bessere Unterstützung bei der Verknüpfung von Anforderungen, d. h. Darstellung der Abhängigkeiten/Auswirkungen von Anforderungsänderungen

- Teil-automatisierte Überführung von unstrukturierten anforderungsrelevanten Informationen in strukturierte Anforderungen

- Annotation und Verknüpfung von Anforderungen mit Hilfe von semantischen Technologien

- Teil-automatisierte Analyse von Anforderungen mit Hilfe von Textmining-Technologien

- Leichtere Weiterverarbeitung der Anforderungen durch Schnittstellen zu externen Tools (z. B. Word, DOORS, Projektmanagement-Tools)

Im Folgenden wird der aktuelle Stand der Forschung im Bereich des wiki-basierten Requirements Engineering vorgestellt.

3 Stand der Forschung

Wikis werden in der Forschung aus verschiedenen Perspektiven betrachtet. Zum einen werden die Erstellungsprozesse von Inhalten in Wikis betrachtet (vgl. z.B. [Dfg06], [PeSe06]). Im Rahmen der Psychologie wird untersucht, was Benutzer eines Wikis zur freiwilligen Mitarbeit motiviert (vgl. [ScHe07]). Zum anderen werden Anwendungsmöglichkeiten von Wikis untersucht, zum Beispiel im Wissensmanagement [Wagn04].

Als mit unseren Arbeiten verwandte Gebiete betrachten wir das kollaborative Requirements Engineering (Abschnitt 3.1) und im das Requirements Engineering mit Wiki-Systemen (Abschnitt 3.2) als spezielle Ausprägung des kollaborativen Arbeiten.

Weiterhin betrachten wir das marktgetriebene Requirements Engineering (Abschnitt 3.3) (engl. Market-Driven Requirements Engineering), da das marktgetriebene Requirements Engineering einen besonderen Fokus auf große Stakeholdergruppen legt.

Abschließend werden wir die verwandten Arbeiten ausgehend von den Zielen des wiki-basierten Requirements Engineering bewerten (Abschnitt 3.4).

3.1 Kollaborative Ansätze für das Requirements Engineering

Zahlreiche kollaborative Ansätze basierend auf Groupwaresystemen wurden für das Requirements Engineering vorgeschlagen. Nachfolgend präsentieren wir einige exemplarische Ansätze:

- Anton et al. [AnLR96] beschreiben ein webbasiertes Werkzeug (GBRAT) für das zielbasierte Requirements Engineering. Das Werkzeug unterstützt die Identifikation, die Verfeinerung und das Management von Zielen zur Spezifikation von Anforderungen.

- Easterbrook und Callahan ([EaCa96]) beschreiben ein Werkzeug (WHERE) zur Unterstützung des kollaborativen Requirements Engineering basierend auf Viewpoints.

- Herlea und Greenberg ([HeGr98]) beschreiben die Verwendung der TeamWave-Plattform zur Unterstützung des Requirements Engineering mit räumlich getrennten Stakeholdern.

- Boehm et al. ([BoGB01]) beschreiben in ihrem Beitrag ihre Erfahrungen bei der Entwicklung und Anwendung von Groupwaresystemen zur verteilten Übereinstimmung von Anforderungen mit dem WinWin-Ansatz.

- Sinha et al. [SiSC06] beschreiben in ihrem Beitrag das Werkzeug EGRET (Eclipse-based Global REquirements Tool) zur Unterstützung des verteilten Anforderungsmanagement.

Die präsentierten Ansätze fokussieren den kollaborativen Aspekt des Requirements Engineering, d.h. die synchrone bzw. asynchrone Zusammenarbeit von räumlich verteilten Stakeholdern während des Requirements Engineering. Es wurde jedoch kein besonderer Fokus auf große Stakeholdergruppen gelegt.

3.2 Einsatz von Wiki-Systemen im Requirements Engineering

Geisser und Hildenbrand ([GeHi06]) beschreiben in ihrem Beitrag eine agile Methode für das verteilte Requirements Engineering und das Änderungsmanagement basierend auf Wiki-Systemen und einer kollaborativen Entwicklungsplattform. Die Autoren argumentieren, dass die präsentierte Methode unter anderem „eine effiziente Miteinbeziehung aller relevanten Interessenvertreter“ ([GeHi06], S. 41) ermöglicht. Die Einbeziehung einer großen Anzahl von Stakeholdern (Interessenvertretern) in das Requirements Engineering wird nicht explizit angestrebt. Probleme, die sich aus einer sehr großen Anzahl von Stakeholdern ergeben können (z.B. redundante oder mehrdeutige Anforderungen, vgl. [LaHa07]) werden nicht diskutiert.

Decker et al. ([DeRR07]) diskutieren in ihrem Beitrag die Herausforderungen an eine Plattform zur Einbeziehung einer großen Anzahl von Stakeholdern, z.B. verschiedene Perspektiven auf das System, verschiedenes Hintergrundwissen, unterschiedliche Fähigkeiten. Ausgehend von diesen Herausforderungen diskutieren Decker et al. Vor- und Nachteile häufig genutzter Requirements-Engineering-Werkzeuge. Die Autoren beschreiben im Weiteren eine mögliche Dokumentenstruktur zur Dokumentation von Anforderungen in Wiki-Systemen. Diese Dokumentenstruktur schließt unter anderem Use Cases und User Stories ein. Die präsentierte Dokumentenstruktur wurde von den Autoren in verschiedenen Projekten mit bis zu 20 Stakeholdern eingesetzt. Die Erfahrungen der Autoren in den verschiedenen Projekten haben gezeigt, dass Wiki-Systeme die Zusammenarbeit von Stakeholdern im Requirements Engineering unterstützen. Eine explizite Aussage über die Skalierbarkeit von Wiki-Systemen für große Stakeholdergruppen fehlt in diesem Beitrag.

3.3 Marktgetriebenes Requirements Engineering

Bei der Entwicklung von Softwaresystemen für einen Massenmarkt (z.B. Textverarbeitungssysteme) findet das marktgetriebene Requirements Engineering Anwendung (vgl. [DaKP03]. Beim marktgetriebenen Requirements Engineering müssen die Anforderungen einer sehr großen Anzahl von Stakeholdern erfasst werden, die mit dem geplanten System arbeiten sollen.

Etablierte Techniken im Requirements Engineering für Einzelsysteme (z.B. Interviews oder Workshops) können im marktgetriebenen Requirements Engineering nur bedingt angewendet werden, da diese Techniken einen sehr großen Zeitaufwand bedeuteten und somit dem Ziel einer schnellen Markeinführung (Time-to-Market, vgl. [SaSK99]) des marktgetrieben Requirements Engineering widersprechen. Stattdessen werden im marktgetriebenen Requirements Engineering Techniken der Marktforschung eingesetzt (vgl. [KeCa95], z.B. Befragungen, Marktstudien), um möglichst repräsentative Aussagen über die Anforderungen der Stakeholder des geplanten Systems zu erhalten.

3.4 Bewertung verwandter Arbeiten

Bisherige kollaborative Ansätze für das Requirements Engineering inklusive der auf Wiki-Systemen beruhenden Ansätze beziehen zwar Stakeholder unmittelbar in das Requirements Engineering mit ein, fokussieren aber nicht explizit die Unterstützung von großen Stakeholdergruppen. Die Betrachtung des marktgetriebenen Requirements Engineering hat gezeigt, dass die Anforderungen großer Stakeholdergruppen mit Hilfe von Techniken der Marktanalyse möglichst repräsentativ erfasst werden. Eine unmittelbare Einbeziehung möglichst vieler Stakeholder in das Requirements Engineering ist nicht explizit vorgesehen.

4 Einbettung des wiki-basierten Requirements Engineering in den Software-Entwicklungsprozess

Software-Entwicklungsprozesse wie z.B. das Wasserfallmodell [Roy87] oder das V-Modell [VMo04] bestehen aus den Phasen: Formulierung von Anforderungen, Design, Implementierung und Test des geplanten Systems. Wiki-basierte Anwendungen unterstützen das Requirements Engineering in der Phase der Formulierung von Anforderungen und liefern damit Informationen für den Design, die Implementierung und das Testen eines geplanten Systems. Abbildung 1 gibt einen groben Überblick über die Einbettung des wiki-basierten Requirements Engineering in den Software-Entwicklungsprozess.

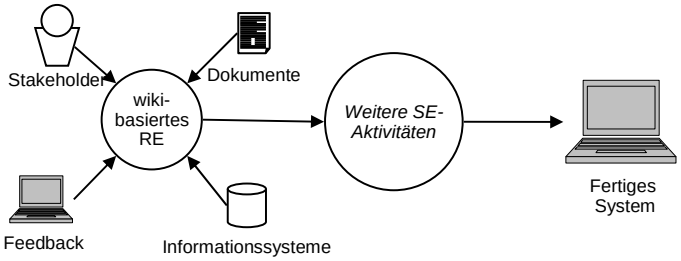


Abbildung 1: Einbettung in den Software-Entwicklungsprozess

Im Folgenden befassen wir uns zunächst mit verschiedenen Anwendungsfällen des wiki-basierten Requirements Engineering im Software-Entwicklungsprozess. Anschließend erläutern wir die Anforderungsquellen für das wiki-basierte Requirements Engineering. Am Ende dieses Kapitels beschreiben wir die Ausgaben des wiki-basierten Requirements Engineering an die weiteren Aktivitäten im Software-Entwicklungs-prozess.

4.1 Anwendungsfälle für das wiki-basierte Requirements Engineering

Wir unterscheiden die Anwendungsfälle für das wiki-basierte Requirements Engineering anhand von zwei Dimensionen. Zum einen betrachten wir die *Projektsituation* in der das wiki-basierte Requirements Engineering angewendet wird. Zum anderen betrachten wir die *Projektziele* für die das wiki-basierte Requirements Engineering zum Einsatz kommt.

Im wiki-basierten Requirements Engineering werden je nach Projektsituation zwei Anwendungsfälle unterschieden:

1. **Wiki-basiertes Requirements Engineering als Ergänzung zum Requirements Engineering**: In diesem Anwendungsfall dient das wiki-basierte Requirements Engineering als Ergänzung für das Requirements Engineering. Die Anforderungen, die während des wiki-basierten Requirements Engineering formuliert und abgestimmt werden, dienen als Grundlage für das Requirements Engineering des geplanten Systems. Das wiki-basierte Requirements Engineering übernimmt bei diesem Anwendungsfall die Rolle eines Integrators für die Ideen und Anforderungen einer großen Stakeholdergruppe. Die gesammelten Anforderungen werden zu einem definierten Zeitpunkt an das Requirements Engineering übergeben und dort mit einer handhabbaren Anzahl von Stakeholdern weiter verfeinert bzw. ergänzt. Diesen Anwendungsfall sehen wir primär als Anwendungsfall für kommerzielle Entwicklungen in Unternehmen.

2. **Wiki-basiertes Requirements Engineering als ausschließliches Requirements Engineering**: In diesem Anwendungsfall dient das wiki-basierte Requirements Engineering als ausschließliche Phase zur Formulierung von Anforderungen, d.h. es folgt keine zusätzliche Phase der Anforderungsformulierung. Im Anwendungsfall nimmt das wiki-basierte Requirements Engineering eine wesentlich zentralere Rolle im Entwicklungsprozess ein, da die formulierten Anforderungen der Stakeholder

direkt in den weiteren Entwicklungsprozess einfließen. Für diesen Anwendungsfall werden wesentlich größere Ansprüche an die Qualität der formulierten Anforderungen gestellt (z.B. Eindeutigkeit, Atomarität, etc.) als im ersten Anwendungsfall, da keine weitere qualitätssichernde Instanz die Anforderungen überprüft. Diesen Anwendungsfall sehen wir primär als Anwendungsfall für nicht-kommerzielle Entwicklung und Open Source sowie kleinere kommerzielle Projekte.

Das wiki-basierte Requirements Engineering unterscheidet je nach Projektkategorie zwei Anwendungsfälle:

1. **Neuentwicklung**: Bei der Neuentwicklung werden Anforderungen an ein neues und bisher noch nicht umgesetztes System formuliert. Als Resultat des wiki-basierten Requirements Engineering werden bei einer Neuentwicklung Anforderungen an das neue System erwartet.

2. **Weiterentwicklung**: Bei der Weiterentwicklung wird ein existierendes System weiterentwickelt bzw. verbessert. Als Resultat des wiki-basierten Requirements Engineering werden bei einer Weiterentwicklung Anforderungen an Verbesserungen für das geplante System erwartet. Diese Anforderungen sollten mehr oder weniger stark mit dem existierenden System gekoppelt sein, um die Verbesserungen möglichst konkret umsetzen zu können.

4.2 Anforderungsquellen für das wiki-basierte Requirements Engineering

Im Rahmen des wiki-basierten Requirements Engineering werden prinzipiell vier Kategorien von Anforderungsquellen unterschieden (siehe auch Abbildung 1):

- **Stakeholder:** Ein Stakeholder ist eine Person oder eine Organisation, die ein potenzielles Interesse an dem zukünftigen System hat und somit in der Regel auch Anforderungen an das System stellt (vgl. [Pohl 2007]).

- **Dokumente**: Unter Dokumenten verstehen wir eine (weitgehend) statische Sammlung von Informationen mit Bezug zum geplanten System (z.B. Benutzerhandbuch, Anforderungsspezifikation, Systemvisionsdokument).

- **Foren/Weblogs**: Hierunter verstehen wir Hilfsmittel mit deren Hilfe Stakeholder eines existierenden Systems über ein existierendes System diskutieren können bzw. ihre Meinung über das existierende System veröffentlichen können. Im Unterschied zu Dokumenten, deren Inhalt statisch ist, können sich Inhalte von solchen Systemen ändern bzw. können erweitert werden.

- **Feedback**: Unter Feedback verstehen wir die Meinungen/Eindrücke von Nutzern zu Bestandteilen eines existierenden Systems. Im Unterschied zu Foren/Weblogs wird das Feedback zu einem existierenden System mit Hilfe des existierenden Systems (oder geeigneter Erweiterungen) gegeben.

Aus der Klassifikation der Anforderungsquellen wird unmittelbar ersichtlich, dass die verfügbaren Anforderungsquellen vom Anwendungsfall bzw. vom Projektziel abhängen. Tabelle 1 zeigt in den Zeilen die Kategorien von Anforderungsquellen und in den Spalten die möglichen Projektziele. Die Einträge der Tabelle enthalten typische Beispiele von Anforderungsquellen, wie sie bei einer Projektziel-Quellenkategorie auftreten können.

Stakeholder sind für beide Projektziele identisch, wobei bei der Weiterentwicklung sicherlich ein stärkerer Fokus auf die Nutzer gelegt wird bzw. bei einer Neuentwicklung potentielle Nutzer in das Requirements Engineering einbezogen werden. Als Dokumente sind bei der Neuentwicklung häufig Ideendokumente, Konzepte, aber auch Gesetzte und Standards verfügbar. Bei einer Weiterentwicklung stehen als Dokumente neben Gesetzen und Standards zum Beispiel Fehlerreports oder Handbücher des existierenden Systems zur Verfügung. Die Anforderungsquelle Feedback ist bei einer Neuentwicklung nicht vorhanden.

Tabelle 1: Beispiele für Anforderungsquellen bei Neu- bzw. Weiterentwicklungen

	Neuentwicklung	**Weiterentwicklung**
Stakeholder	(potenzielle) Nutzer, Techniker, Entwickler, Datenschutzbeauftragte	
Dokumente	Ideendokumente, Konzepte, Gesetzte, Standards	Fehlerreports, Gesetzte, Standards
Feedback aus existierenden Systemen	*Nicht vorhanden*	Feedback von Nutzern des laufenden Systems
Foren/Weblogs	Projektrelevante Weblogs und Diskussionsforen	Weblogs, Diskussionsforen über das laufende System

4.3 Ausgabe an weitere Aktivitäten im Software-Entwicklungsprozess

Abhängig von der Projektsituation werden am Ende des wiki-basierten Requirements Engineering die Anforderungen entweder an das Requirements Engineering oder an weitere Software-Entwicklungsaktivitäten (z.B. Entwurf der Architektur) übergeben. Die genaue Ausprägung der Anforderungen hängt sowohl vom Projektziel als auch von der Projektsituation ab.

Abhängig vom Projektziel müssen die Anforderungen mit Bezugspunkten versehen werden. Für beide Typen von Projektzielen (Neu- und Weiterentwicklung) müssen die Anforderungen mit ihren jeweiligen Anforderungsquellen verknüpft werden, um ggf. zusätzliche Informationen zu den Anforderungen erfragen zu können. Bei der Weiterentwicklung beinhalten die Anforderungen weitestgehend Verbesserungen des existierenden Systems. Dementsprechend sollten die Anforderungen zusätzlich mit den

jeweiligen Bezugspunkten des existierenden Systems verknüpft werden, deren Verbesserung durch die Anforderungen beschrieben wird. Wenn zum Beispiel eine Anforderung beinhaltet, dass ein bestehender E-Mailclient neben dem PO3-Protokoll auch das IMAP-Protokoll unterstützen soll, dann sollte für diese Anforderung vermerkt werden, dass sich diese Anforderung auf die Kommunikationsprotokolle des E-Mailclients bezieht.

Abhängig von der Projektsituation (wiki-basiertes Requirements Engineering als Ergänzung oder ausschließlich) sollten die Anforderungen unterschiedlich stark detailliert sein. Wird das wiki-basierte Requirements Engineering als ausschließliches Requirements Engineering angewendet, so müssen die Anforderungen wesentlich detaillierter sein. Die Anforderungen müssen soviel Informationen enthalten, dass nachfolgende Software-Entwicklungsaktivitäten basierend auf diesen Anforderungen das geplante System entwickeln können bzw. Verbesserungen an existierenden Systemen vornehmen können. Der Umkehrschluss soll an dieser Stelle nicht gelten. Bei einer Neuentwicklung können ebenfalls sehr detaillierte und qualitativ hochwertige Anforderungen formuliert werden, es ist jedoch nicht zwingend erforderlich.

Neben der Projektsituation und dem Projektziel bestimmen die verfügbaren Werkzeuge die Ausgabe des wiki-basierten Requirements Engineering. Das Tool für das wiki-basierte Requirements Engineering muss die Anforderungen in dem einen Datenformat zur Verfügung stellen, dass von den Werkzeugen verstanden wird, die in den nachgelagerten Software-Entwicklungsaktivitäten verwendet werden. Dieser Aspekt des wiki-basierten Requirements Engineering wird in diesem Beitrag jedoch nicht weiter behandelt, da es sich hierbei um technische Details handelt.

5 Der Prozess des wiki-basierten Requirements Engineering im Überblick

Der Prozess des wiki-basierten Requirements Engineering umfasst drei Teilprozesse: „Informationen analysieren und zuordnen“, „Anforderungen gewinnen und übereinstimmen“ und „Anforderungen aufbereiten“ (s. Abbildung 2). Diese Prozesse und deren Verknüpfungen werden in den nachfolgenden Abschnitten erläutert. Dabei wird jeder Prozess einheitlich hinsichtlich Ein- und Ausgaben und der Verarbeitung der Informationen beschrieben. Ferner wird darauf hingewiesen, wie Semantic-Web-Technologien diese Prozesse unterstützen können.

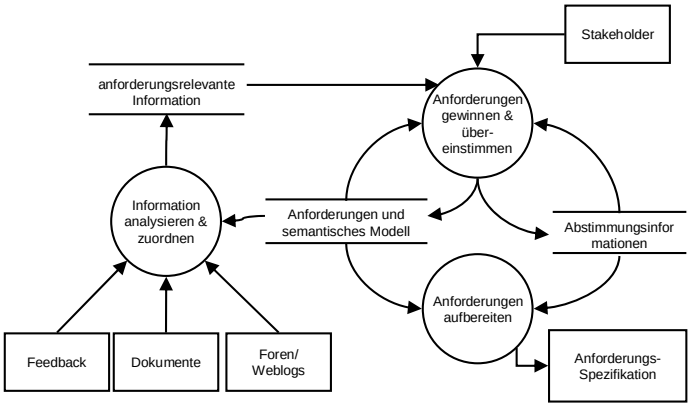


Abbildung 2: Überblick über das Prozessmodell

5.1 Prozess „Anforderungen gewinnen und übereinstimmen“

Der Prozess hat zum Ziel, Anforderungen zu dokumentieren bzw. zu modifizieren sowie bezüglich der Anforderungen eine Mehrheitsmeinung herzustellen.

Informationsquellen (Eingaben) für den Prozess: Stakeholder, bereits bestehende Anforderungen und deren semantische Annotationen (Feedback, zusätzliche Informationen, Forumsbeiträge), Abstimmungsinformationen.

Ausgaben des Prozesses: Ergebnis des Prozesses sind neue oder überarbeitete Anforderungen bzw. Abstimmungsinformationen.

Verarbeitung der Informationen innerhalb des Prozesses: Der Stakeholder dient als wichtigste Datenquelle für die Anforderungsgewinnung. Ferner werden bereits bestehende Anforderungen und deren semantische Annotationen (Feedback, zusätzliche Informationen, Forumsbeiträge) verwendet, um Anforderungen zu erstellen oder zu bearbeiten. Darüber hinaus werden Abstimmungsinformationen für die Anforderungserstellung verwendet.

Der Prozess besteht aus drei Schritten: der Anforderungserstellung, der Anforderungsbearbeitung und der Anforderungsübereinstimmung. Ergebnis des Prozesses sind neue oder überarbeitete Anforderungen sowie ggf. Änderungen am SWREO. Ferner werden Abstimmungsinformationen erzeugt.

Nutzen von Semantic-Web-Technologien: Parallel zur Eingabe von Anforderungen können Verweise auf ähnliche Inhalte oder Duplikate sowie Informationen über relevante Daten wie z.B. Autor, Anzahl Stakeholder, Abstimmungszahlen erfolgen. Ebenso kann eine gezielte Suche nach Anforderungen unterstützt werden.

5.1.1 Detaillierung des Prozesses „Anforderungen gewinnen und übereinstimmen“

Der Prozess „Anforderungen gewinnen und übereinstimmen“ ist neben den Prozessen „Informationen analysieren und zuordnen“ und „Anforderungen aufbereiten“ ein wesentlicher Kern des wiki-basierten RE. Abbildung 3 zeigt die zwei Teilprozesse des Prozesses „Anforderungen gewinnen und übereinstimmen“:

- *„Anforderungen gewinnen und bearbeiten“*: In diesem Teilprozess werden die Anforderungen der Stakeholder gewonnen bzw. bestehende Anforderungen durch Stakeholder überarbeitet. Dieser Teilprozess wird in Abschnitt 5.1.1.1 weiter beschrieben.

- *„Anforderungen übereinstimmen“*: In diesem Teilprozess werden die Anforderungen unter den Stakeholdern abgestimmt und diskutiert. Dieser Teilprozess wird in Abschnitt 5.1.1.2 weiter beschrieben.

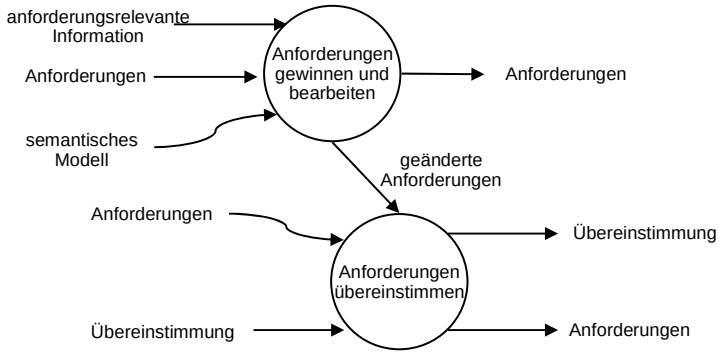


Abbildung 3: Prozess „Anforderungen gewinnen und übereinstimmen“

5.1.1.1 „Anforderungen gewinnen und bearbeiten“

Die Gewinnung bisher nicht betrachteter Anforderungen bzw. die Bearbeitung bereits bestehender Anforderungen ist eine Kernaufgabe aller Stakeholder im wiki-basierten RE. Die Vielzahl von Stakeholdern bringt ihre Ideen über das geplante System in das Wiki ein bzw. bearbeitet bestehende Anforderungen gemäß seinen Vorstellungen. Neben den eigentlichen Anforderungsartefakten können Stakeholder auch Beziehungen zwischen bestehenden Anforderungen definieren, um zum Beispiel Abhängigkeiten oder Gemeinsamkeiten auszudrücken. Auf diesem Wege entsteht zwischen den Stakeholdern durch das Erstellen und Bearbeiten von Anforderungen ein kreativer Dialog über das geplante System. Neben den eigentlichen Anforderungen bietet das wiki-basierte RE den Stakeholdern zusätzliche anforderungsrelevante Informationen, die zuvor aus anderen Quellen extrahiert wurden.

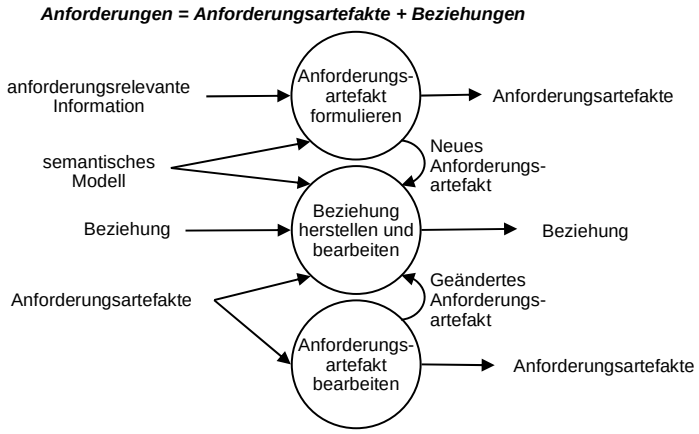


Abbildung 4: Prozess „Anforderungen erstellen und bearbeiten“

Informationsquellen (Eingaben) für den Prozess: Anforderungsrelevante Informationen, semantisches Modell und Anforderungsartefakte. Die Informationsquelle „Anforderungen“ aus dem Prozess „Anforderungen gewinnen und übereinstimmen“ wird in den Informationsquellen „Anforderungsartefakte“ und „Beziehungen“ verfeinert.

Ausgaben des Prozesses: Anforderungsartefakte (neue oder geänderte), Beziehungen zwischen Artefakten.

Verarbeitung der Informationen innerhalb des Prozesses: a) Stakeholder können im Prozess „Anforderungsartefakt formulieren“ ein neues Anforderungsartefakt in das System eingeben. Diese Eingaben werden analysiert. Dem Stakeholder werden daraufhin relevante Informationen aus den anforderungsrelevanten Informationen für die Formulierung von Anforderungen zur Verfügung gestellt. Diese Informationen werden u. a. dazu genutzt, um Anforderungen zu detaillieren.

b) Im Prozess „Anforderungsartefakt bearbeiten“ werden bestehende Anforderungsartefakte durch die Stakeholder angepasst. Gemäß der Wiki-Philosophie werden sämtliche Änderungen an den Anforderungen nachvollziehbar dokumentiert, sodass eine Änderung zu jeder Zeit rückgängig gemacht werden kann bzw. geprüft werden kann, wie sich eine Anforderung über den Verlauf der Zeit geändert hat. Diese Nachvollziehbarkeit sämtlicher Änderungen soll bei den Stakeholdern die Hemmschwelle abbauen, bestehende Artefakte anzupassen, wenn die Artefakte nicht ihren Vorstellungen entsprechen.

c) Neue und auch geänderte Anforderungen sollen mit bestehenden Artefakten in Beziehung gesetzt werden. Dies ist im Prozess „Beziehungen herstellen“ möglich. Wir unterscheiden drei Möglichkeiten für den Prozess „Beziehungen herstellen“:

- Bei Möglichkeit 1 kann der Stakeholder unmittelbar Beziehungen zwischen bestehenden Artefakten herstellen bzw. anpassen.

- Bei Möglichkeit 2 wird der Stakeholder nach dem Erstellen eines neuen Artefaktes dazu aufgefordert, Beziehungen vom neuen Artefakt zu bestehenden Artefakten herzustellen.

- Bei Möglichkeit 3 wird der Stakeholder nach der Anpassung eines Artefaktes dazu aufgefordert, die bestehenden Beziehungen zu überprüfen bzw. neue Beziehungen, die sich aus den Änderungen am Artefakt ergeben haben können, herzustellen.

Nutzen von Semantic Web-Technologien: Semantic Web-Technologien unterstützen in diesem Prozess vor allem die Zuordnung von Anforderungen zu anforderungsrelevanten Informationen.

5.1.1.2 „Anforderungen übereinstimmen“

Der Prozess „Anforderungen gewinnen und bearbeiten“ fokussiert die Gewinnung neuer bzw. die Änderung von bestehenden Anforderungsartefakten. Neue Artefakte bzw. Änderungen an bestehenden Artefakten müssen nicht unbedingt von allen Stakeholdern getragen werden. Der Prozess „Anforderungen übereinstimmen“ hat die Aufgabe, die bestehenden Anforderungsartefakte (inkl. Änderungen) durch die Stakeholder abstimmen zu lassen, d.h. ein Stakeholder sollte nach Möglichkeit zu jedem Artefakt seine Zustimmung bzw. seine Ablehnung erklären. Werden Anforderungsartefakte von einigen Stakeholdern abgelehnt, so entsteht ein Konflikt zwischen den Stakeholdern, die dem Artefakt zustimmen und den Stakeholdern, die das Artefakt ablehnen. Dieser Konflikt sollte mit dem Ziel der Einigung unter den Stakeholdern diskutiert werden. Die Durchführung von Diskussionen ist ebenfalls Aufgabe des Prozesses „Anforderungen übereinstimmen“.

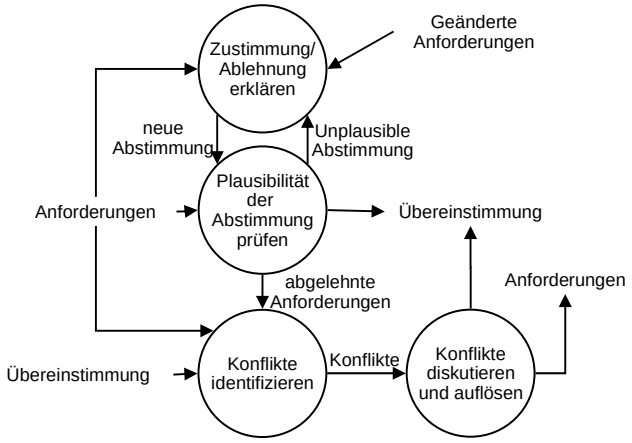


Abbildung 5: Prozess „Anforderungen übereinstimmen“

Informationsquellen (Eingaben) für den Prozess: Anforderungen, Übereinstimmungsinformation.

Ausgaben des Prozesses: Änderungen an den Anforderungen (basierend auf Diskussionsergebnissen), Übereinstimmungsinformation.

Verarbeitung der Informationen innerhalb des Prozesses: Der Stakeholder erklärt im Prozess „Zustimmung/Ablehnung erklären“ für bestehende Anforderungen seine Zustimmung bzw. Ablehnung. Als Standard für jede Anforderung ist bei jedem Stakeholder die Abstimmung „Neutral“ definiert, d.h. wenn ein Stakeholder nicht explizit einer Anforderung zustimmt oder sie ablehnt, kann keine Aussage über seine Abstimmung gemacht werden und das System geht von Neutralität aus. Eine Ausnahme von diesem Standard besteht für den Stakeholder der eine Anforderung eingibt bzw. ändert. In diesem Fall wird die Abstimmung des Stakeholders auf Zustimmung gesetzt, da implizit davon ausgegangen wird, dass ein Stakeholder seiner eigenen Anforderung bzw. Änderung zustimmt.

Ändert sich eine Anforderung über die ein Stakeholder bereits abgestimmt hat, so wird er im Prozess „Zustimmung/Ablehnung erklären“ erneut aufgefordert, seine Zustimmung bzw. Ablehnung dahingehend zu überprüfen, ob die Änderung Einfluss auf seine Abstimmung hatte.

Mit Hilfe von Beziehungen zwischen Anforderungsartefakten können Plausibilitätsprüfungen zwischen den Abstimmungen eines Stakeholders für verschiedene Anforderungsartefakte vorgenommen werden. Wenn zum Beispiel zwei Anforderungen durch eine „Bedingt“-Beziehung miteinander verbunden sind und ein Stakeholder einer Anforderung zustimmt und die andere Anforderung ablehnt, so kann dies auf einen Widerspruch hindeuten. Dieser Widerspruch wird vom System angezeigt und der Stakeholder wird aufgefordert, seine Abstimmung erneut zu überprüfen. Wenn eine Abstimmung als plausibel eingestuft wurde, wird sie in den Übereinstimmungsinformationen abgelegt.

Ausgehend von bereits abgestimmten Anforderungen (Übereinstimmungsinformation) und neuen Abstimmungen ermittelt der Prozess „Konflikte identifizieren“ Konflikte zwischen Stakeholdern.

Identifizierte Konflikte werden durch die Stakeholder im Prozess „Konflikte diskutieren und auflösen“ diskutiert. Eine Diskussion sollte immer mit dem Ziel geführt werden, die Konfliktparteien zu einigen, z.B. davon zu überzeugen, die Abstimmung zu ändern oder als Konsens die konfliktären Anforderung zu ändern.

Nutzen von Semantic-Web-Technologien: Die Menge an Anforderungen, über die Stakeholder abstimmen können bzw. die von Stakeholder diskutiert werden können, ist beim wiki-basierten RE sehr groß. Daher kann es möglich sein, dass ein einzelner Stakeholder nicht über alle Anforderungen abstimmen kann bzw. sich an allen Diskussionen mit gleicher Intensität beteiligen kann. Semantic-Web-Technologien werden in diesem Fall dazu genutzt, dem Stakeholder einen strukturierten Zugang zur Abstimmung und zur Diskussion von Anforderungen zu bieten. Denkbar ist zum Beispiel, dass dem Stakeholder im Rahmen des wiki-basierten RE Anforderungen (oder auch Diskussionen) vorgeschlagen werden, die basierend auf seinen bisherigen Aktivitäten (z.B. eingegebene Anforderungen oder Beteiligungen an Diskussionen) für ihn interessant sein könnten.

5.2 Prozess “Anforderungen aufbereiten”

Der Prozess hat zum Ziel, Anforderungen so zu transformieren, dass diese in kommerzielle RE-Werkzeuge (z.B. Doors, Requisite Pro) importiert und weiterverarbeitet werden können.

Informationsquellen (Eingaben) für den Prozess: Neue oder überarbeitete Anforderungen, Abstimmungsinformationen (wozu braucht man die Informationen?)

Ausgaben des Prozesses: Die transformierten Anforderungen werden kommerziellen RE-Werkzeugen zur Verfügung gestellt.

Verarbeitung der Informationen innerhalb des Prozesses: Es findet eine Abbildung der Attribute einer Wiki-Anforderung auf die entsprechenden Attribute statt. Hierbei können Attributwerte verworfen, vereinigt oder aufgeteilt werden.

Nutzen von Semantic-Web-Technologien: Berücksichtigung/Erhaltung der semantischen Struktur bei der Aufbereitung der Anforderungen, z.B. bei Tools mit einem eigenen Informationsmodell.

5.3 Prozess “Informationen analysieren und zuordnen”

Der Prozess hat zum Ziel, Informationen wie Nutzerfeedback, systemrelevante Dokumente und Internetforen automatisiert zu analysieren und anforderungsrelevante Inhalte zu identifizieren. Eine ausführliche Beschreibung des Prozesses findet sich in [CLVZ07] in diesem Tagungsband.

6 Zusammenfassung und Ausblick auf geplante Forschungsaktivitäten

Dieser Beitrag beschäftigt sich mit der Frage, wie Wiki-Systeme ergänzt durch Semantic-Web-Technologien das Requirements Engineering für große und verteilte Benutzergruppen unterstützen können. Das Requirements Engineering mit solchen Benutzergruppen stellt eine besondere Herausforderung dar. Durch die große Anzahl und räumliche Verteilung der Stakeholder ist der Aufwand für Befragung, Ermittlung und Abstimmung von Anforderungen sehr hoch. Diese Annahme wurde im Rahmen einer Unternehmensstudie des Projekts SoftWiki bestätigt.

Im Rahmen des Projektes SoftWiki wurde das wiki-basierte Requirements Engineering als Lösungsansatz entwickelt, um Requirements Engineering mit großen und verteilten Stakeholdergruppen zu betreiben. In diesem Beitrag wurde das wiki-basierte Requirements Engineering vorgestellt. Dabei wurde darauf eingegangen, wie Semantic-Web-Technologien nutzenbringend eingesetzt werden können. Ferner wurde erläutert, wie das wiki-basierte Requirements Engineering in die üblichen Softwaretechnik-

Aktivitäten wie herkömmliches Requirements Engineering, Entwurf, Implementierung usw. eingefügt wird.

Dieser Beitrag hat gezeigt, dass das wiki-basierte Requirements Engineering eine mögliche Lösung für den Umgang mit großen und verteilten Stakeholdergruppen ist. Möglich wird diese Lösung durch die Kombination der Konzepte Semantic Web, Social Software und Knowledge Management. Das wiki-basierte Requirements Engineering nutzt die Synergien aus allen drei Konzepten, um das Requirements Engineering mit großen und verteilten Stakeholdern zu unterstützen (siehe Abbildung 6).

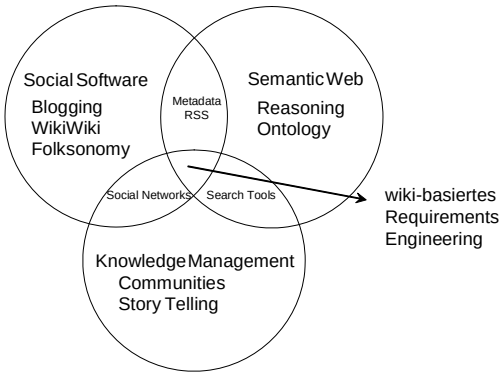


Abbildung 6: Einordnung von SoftWiki in aktuelle Konzepte (in Anlehnung an [Sem07])

Im Rahmen des SoftWiki-Projektes ist es geplant, den Nutzen von Wiki-Systemen im Requirements Engineering durch Experimente und Fallstudien zu untersuchen. In einer Pilotphase werden generische Wiki-Systeme in studentischen Projekten zur Unterstützung des Requirements Engineering eingesetzt. Die Organisation der Wiki-Inhalte und die Durchführung des Requirements Engineering werden dabei nicht durch das Wiki-System vorstrukturiert und bleiben den Studierenden überlassen.

In einer zweiten Pilotphase wird das SoftWiki-System in studentischen Projekten eingesetzt, welches den oben beschriebenen SoftWiki-Requirements-Engineering-Prozess unterstützt durch Textmining und Semantic-Web-Technolgien umsetzt.

Der Vergleich zwischen der ersten und zweiten Pilotphase soll zeigen, ob und in wie weit ein strukturierter Prozess unterstützt durch Textmining und Semantic-Web-Technologien einem generischen Wiki-System überlegen ist.

Die Erfahrungen aus den studentischen Projekten dienen dazu, den initialen Prototypen des Softwiki-Systems zu verbessern. Das verbesserte Softwiki-System soll schließlich in industriellen Pilotprojekten eingesetzt werden, um den Nutzen in einem industriellen Umfeld zu untersuchen.

7 Literaturverzeichnis

[AnLR96] Anton, A., Liang, E, Rodenstein, R.A.: A web-based requirements analysis tool. In: Proceedings of the Fifth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'96), 1996.

[AuRF06] Auer, S.; Riechert, T.; Fähnrich, K.-P.: SoftWiki - Agiles Requirements-Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder. GeNeMe'06 - Gemeinschaft in neuen Medien, 2006.

[Bmbf07] BMBF-Projekt SoftWiki, Förderkennzeichen 01ISF02B, www.softwiki.de, Abruf 09.03.07

[BoGB01] Boehm, B., Grünbacher, P., and Briggs, R. O.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software Vol. 18, No.3, 2001.

[Boeh06] Boehm, B.: A View of 20th and 21st Century Software Engineering. In Proceedings of ICSE'06, May 20–28, 2006, S. 12-29.

[CLVZ07] Cyriaks, H.; Lohmann, S.; Velioglu, V.; Ziegler, J.: Semantische Aufbereitung von Dokumentenbeständen zur Gewinnung anforderungsrelevanter Informationen. In: SoftWiki Project Workshop, 2007.

[DaKP03] Dahlstedt, G., Karlsson, L., Persson, A., Natt och Dag, J., Regnell, B.: Market-Driven Requirements Engineering Processes for Software Products - a Report on Current Practices, RECOTS'03 - Proceedings of the International Workshop on COTS and Product Software, Monterey Bay, California, USA, 2003

[DeRR07] Decker, B.; Ras, R.; Rech, J.; Jaubert, P.; Rieth, M.: Wiki-Based Stakeholder Participation in Requirements Engineering. IEEE Software, Vol 24, No. 2, 2007, S. 29-35.

[Deni07] http://www.denis.bund.de/ueber_denis/index.html, Abruf 09.03.07

[Dfg06] „Konstitution und Erhalt von Kooperation am Beispiel Wikipedia", DFG-Projekt, 2006

[EaCa96] Easterbrook, S.M., Callahan, J.:Independent validation of specifications: a coordination headache. In: Proceedings of the Fifth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'96), 1996

[GeHi06] Geisser, M.; Hildenbrand, T.: Agiles, verteiltes Requirements Engineering mit Wikis und einer Kollaborativen Softwareentwicklungsplattform, Objektspektrum, Nr. 6, 2006

[HaJS06] Happ, S., Jungmann, B., Schönefeld, F.: Web 2.0: Paradigmenwechsel in der Unternehmenskommunikation. In Meißner, K., Engelin, M. (Hrsg.): Virtuelle Organisation und Neue Medien, 2006.

[HeGr98] Herlea, D.; Greenberg, S.: Using a groupware space for distrib-uted requirements engineering. In Proceedings of the Seventh IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, S. 57-62, 1998.

[HiWi05] Hippner, H. & Wilde T., 2005: Social Software. In: Wirtschaftsinformatik, 47 Jg., Nr. 6, S. 441-444. Wiesbaden: Vieweg.

[Katz05] Katz, A., 2006: The Evolving Web. http://web2.toptensources.com/ TopTenSources/Default.aspx, Abruf am 19.04.2006

[KeCa95] Keil, M. and Carmel, E. 1995. Customer-developer links in software development. Commun. ACM Nr. 38, 5, 1995, S. 33-44.

[LaHa07] Lauenroth, K.; Halmans, G.: Auswirkungen sehr vieler Stakeholder auf das Requirements Engineering, Software Technik Trends, Februar 2007

[LeCu01] Leuf, Bo; Cunningham, W.: The Wiki Way: Collaboration and Sharing on the Internet. Addison-Wesley Professional, 2001.

[ORei05a] O'Reilly, T., 2005: What Is Web 2.0. Abruf am 14.04.2006, http://www.oreillynet.com /pub/a/OReilly/tim/news/2005/09/30/what-is-web-20.html?page=1

[ORei05b] O'Reilly, T., 2005: Web 2.0: Compact Definition

[PeSe06] Petzold, C.; Seidenglanz, S.: Foucalt@Wiki – First Steps Towards a Conceptual Framework for the Analysis of Wiki Discourses. In: WikiSym 06, 2006, S. 59-68.

[Rhei94] Rheingold, H.: "Virtuelle Gemeinschaft", Bonn 1994.
[SaSK99] Sawyer, P., Sommerville, I. and Kotonya, G.: Improving Market-Driven RE Processes. In: Proceedings of the International Conference on Product Focuesed Software Process Improvement, Oulu, Finland, 1999.
[ScHe07] Schroer, J.; Hertel, G.: Voluntary Engagement in an Open Web-based Encyclopedia: Wikipedians, and Why They Do It, 2007 Verfügbar unter http://www.abo.psychologie.uni-wuerzburg.de/virtualcollaboration/publications.php?action=view&id=44
[Sing06] Singel, R., 2006: Are You Ready for Web 2.0? http://www.wired.com/news/technology/0,1282,69114,00.html, Abruf 06.10.06
[SiSC06] Sinha, V; Sengupta, B.; Chandra, S.: Enabling Collaboration in Distributed Requirements Management. In: IEEE Software Vol. 23, No. 5, 2006.
[Wagn04] Wagner, C.: Wiki: A Technology for Conversational Knowledge Management and Group Collaboration. In: Communications of the Association for Information Systems, Vol. 13, 2004, S. 265-289.
[Roy87] W. W. Royce: Managing the Development of Large Software Systems. In: Proceedings of the 9th International Conference on Software Engineering (ICSE'87), IEEE Computer Society Press, Los Alamitos, 1987, S. 328-338.
[VMo04] V-Modell® XT – Grundlagen des V-Modells, 2004. Verfügbar unter: http://ftp.uni-kl.de/pub/v-modell-xt/Release-1.1/Dokumentation/pdf/V-Modell-XT-Komplett.pdf; abgerufen am 22.05.2007.
[Poh07] K. Pohl: Requirements Engineering – Grundlagen, Prinzipien und Techniken. Dpunkt.Verlag, 2007.
[Sem07] Semantic Web School. www.semantic-web.at, 2007.

Semantische Aufbereitung von Dokumentenbeständen zur Gewinnung anforderungsrelevanter Informationen

Haiko Cyriaks[1], Steffen Lohmann[2], Horst Stolz[1], Veli Velioglu[1], Jürgen Ziegler[2]

[1]ISA Informationssysteme GmbH
Azenbergstraße 35, 70174 Stuttgart
{cyriaks, stolz, velioglu}@isa.de

[2]Universität Duisburg-Essen
Abt. Informatik und Angew. Kognitionswissenschaft
Lotharstraße 65, 47057 Duisburg
{lohmann, ziegler}@interactivesystems.info

Abstract: Bei der umfassenden Erhebung von Anforderungen muss häufig eine Vielzahl bereits vorhandener Dokumente berücksichtigt werden. Die Aufbereitung und Auswertung dieser Dokumente kann mit einem hohen Aufwand verbunden sein. Um diese Aktivitäten zu erleichtern, werden im SoftWiki-Ansatz Text Mining-Verfahren eingesetzt, die mittels statistischer und korpuslinguistischer Analysen Dokumentenbestände automatisiert vorverarbeiten. Es werden Worthäufigkeiten berechnet und statistisch signifikante Nachbarschafts- und Satz-Kookkurrenzen identifiziert. Das Ergebnis wird als RDF-Graph ausgegeben und in Form eines semantischen Netzes visualisiert. Hierdurch werden ein thematischer Überblick über den Dokumentenbestand und ein leichterer Zugriff auf Teile davon ermöglicht. Die Visualisierung und aktive Filtermöglichkeiten unterstützen die Identifizierung von anforderungsrelevanten Informationen.

1 Einleitung

Im SoftWiki-Projekt[1] verfolgen wir einen integrierten Requirements Engineering-Ansatz, der bei der Anforderungserhebung unterschiedlichste Informationsquellen berücksichtigt: Neben der direkten Beteiligung von Stakeholdern in der kollaborativen SoftWiki-Umgebung sollen anforderungsrelevante Informationen unter anderem möglichst umfassend auch aus bereits existierenden Dokumentenbeständen wie z.B. Anwendungsfall- und Systembeschreibungen oder Kunden-E-Mails gewonnen werden (vgl. auch [Ha07]). Diese Dokumentenbestände weisen in ihrer Gesamtheit jedoch meist nur einen geringen Strukturierungsgrad und eine große thematische Vielfalt auf. Eine Vorverarbeitung dieser Informationen ist deshalb unabdingbare Voraussetzung für ihre effiziente Integration in die kollaborative SoftWiki-Umgebung. Um den Aufwand der Vorverarbeitung zu reduzieren, sollen die Dokumentenbestände automatisch analysiert und semantisch aufbereitet werden.

[1] http://softwiki.de

Vor diesem Hintergrund werden im SoftWiki-Kontext Text Mining-Verfahren [FS06, HQW06] eingesetzt und weiterentwickelt, die aus Dokumentenbeständen zentrale Begriffe und semantische Beziehungen ermitteln und mit Annotationen versehen. Die im SoftWiki Requirements Engineering Prozess angewandten Verarbeitungsschritte Kollokationsanalyse, RDF-Transformation, Visualisierung und manuelle Bearbeitung sowie die weitere Verwendung der semantischen Struktur werden im Folgenden näher erläutert.

2 Kollokationsanalyse und RDF-Transformation

Eine Kernaktivität im Text Mining ist die Identifizierung von Kollokationen im untersuchten Dokumentenbestand. Der Begriff Kollokation bezeichnet das überdurchschnittlich häufige, gemeinsame Auftreten der gleichen Wörter in einem begrenzten Kontext (Kookurrenz) und ist Indikator für eine semantische Beziehung zwischen diesen Wörtern. Der betrachtete Kontext kann dabei variieren. Im SoftWiki-Ansatz werden alle Kookkurrenzen innerhalb von Sätzen sowie zusätzlich die direkten linken und rechten Nachbarn eines Wortes für die Analyse herangezogen.

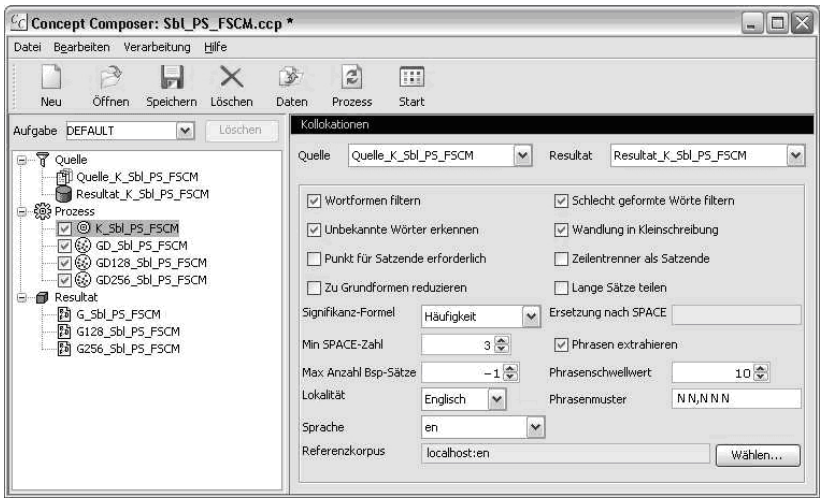


Abbildung 1: ConceptComposer – Parameter für die Kollokationsanalyse

Die Analyse wird mit dem *ConceptComposer* durchgeführt, der zunächst alle Wortformen isoliert, auf ihre Grundformen reduziert und die Häufigkeit ihres Auftretens berechnet. Anschließend werden die Nachbarschafts- und Satzkookkurrenzen ermittelt und mit einem Referenzkorpus abgeglichen. Dieser stellt übliche Häufigkeits- und Kookkurrenzwerte bereit, die durch eine Analyse von etwa zehn Millionen Sätzen ermittelt wurden [HQW02]. Durch den Abgleich mit dem Referenzkorpus lassen sich signifikante Kookkurrenzen identifizieren. Das Ergebnis der Analyse wird in einer relationalen Datenbank abgelegt und bildet die Grundlage für weitere Bearbeitungsschritte.

Über die Benutzeroberfläche des ConceptComposer lässt sich die Kollokationsanalyse anhand verschiedener Parameter konfigurieren (siehe Abbildung 1). Beispielsweise kann definiert werden, wie viele Wörter (bzw. Leerzeichen) ein Satz mindestens enthalten muss, damit er in die Analyse einfließt. Zusätzlich lassen sich Phrasen festlegen, die bei der Analyse berücksichtigt werden sollen. Phrasen bestehen aus mehreren Wörtern und können als Muster (in Form einer Liste von aufeinander folgenden Wortarten) angegeben werden. Treten die gleichen Wörter hinreichend oft entsprechend dem definierten Muster auf, werden sie als Phrase erkannt und extrahiert.

Anschließend werden die ermittelten Begriffe und semantischen Relationen über den Graph Distiller in RDF [Mc04, LS99] transformiert. Abbildung 2 zeigt die Benutzeroberfläche, die diesen Transformationsprozess steuert. Hier lässt sich beispielsweise festlegen, welchen minimalen Signifikanzwert Kollokationen besitzen müssen, damit sie in die RDF-Struktur übernommen werden. Außerdem kann ein Wert für die Anzahl an semantischen Relationen angegeben werden, die eine Wortform in der RDF-Struktur maximal besitzen darf.

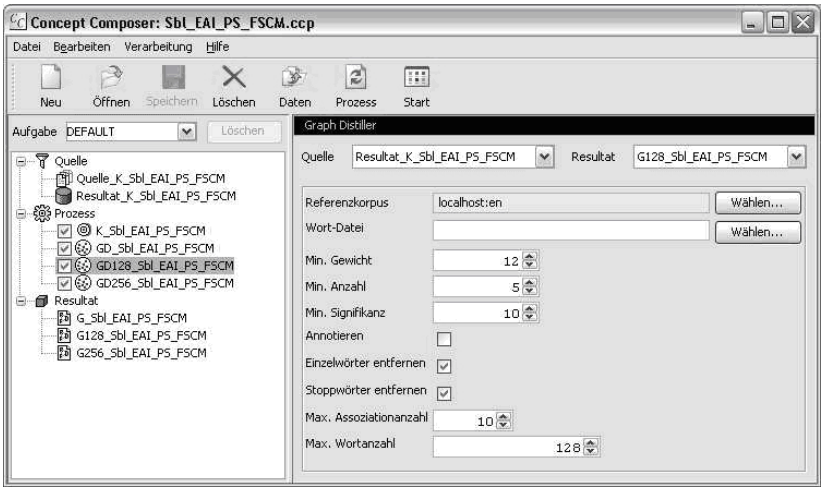


Abbildung 2: ConceptComposer – Parameter für den Graph Distiller

3 Visualisierung

Im nächsten Schritt wird die aus den Quelldokumenten generierte RDF-Struktur in Form eines semantischen Netzes visualisiert. Semantische Netze sind ein beliebtes Mittel der Wissensrepräsentation [ST05, Gr82]. Zentrale Begriffe der betrachteten Domäne werden als Knoten und Beziehungen zwischen diesen Begriffen als Kanten abgebildet. Die Knoten können um Attribute ergänzt werden. Diese netzförmige Verbindung von Begriffen entspricht dem menschlichen Assoziationsdenken und kann einen intuitiven Zugang zu umfangreichen und komplexen Themengebieten darstellen [So91]. Abbildung 3 zeigt ein

semantisches Netz in der Überblicksdarstellung, das aus einem Dokument des SoftWiki-Projektpartners Lecos GmbH generiert wurde, in dem ein spezifischer Anwendungsfall beschrieben wird.

Abbildung 3: Semantisches Netz einer Anwendungsfallbeschreibung

Die Visualisierung des semantischen Netzes auf Basis der zuvor erzeugten RDF-Struktur geschieht mittels *SemanticTalk*. Die extrahierten Wörter werden als Knoten und die ermittelten Kollokationen zwischen den Wörtern als Kanten dargestellt. Die Berechnung der Netzdarstellung basiert auf einem Simulated Annealing Algorithmus [KGV83]. Nach der initialen Generierung des Netzes lässt sich die Anordnung der Knoten manuell beliebig verändern. Bei jedem Knoten ist hinterlegt, an welchen Stellen innerhalb der analysierten Quelldokumente das jeweilige Wort vorkommt.

Abbildung 4 zeigt die Benutzeroberfläche von SemanticTalk. Im rechten Bereich wird das semantische Netz angezeigt, das in diesem Fall aus der Beschreibung eines Customer Relationship Management (CRM)-Systems generiert wurde. Der dargestellte Bereich lässt sich zoomen und verschieben, so dass verschiedene Teilausschnitte wie auch das gesamte Netz betrachtet werden können. Knoten und Kanten können editiert und entfernt werden. Außerdem lassen sich beliebige weitere Knoten, Kanten und Attributehinzufügen. Im linken Bereich der Benutzeroberfläche ist der Graph in einer Miniaturansicht abgebildet, die bei sehr großen Netzen die Übersicht und Orientierung erleichtert. Darüber befinden sich mehrere Regler, mit denen die Ansicht des Graphen angepasst werden kann. Neben dem Zoomfaktor lässt sich hierüber beispielsweise bestimmen, wie viele Kanten des Netzes dargestellt werden sollen. Wird die Anzahl mit dem Regler verringert, werden Kollokationen mit entsprechend geringeren Signifikanzwerten ausgeblendet. Darüber hinaus lassen sich die Knoten nach ihrem Typ filtern, zum Beispiel auf Grundlage der ermittelten Wortarten.

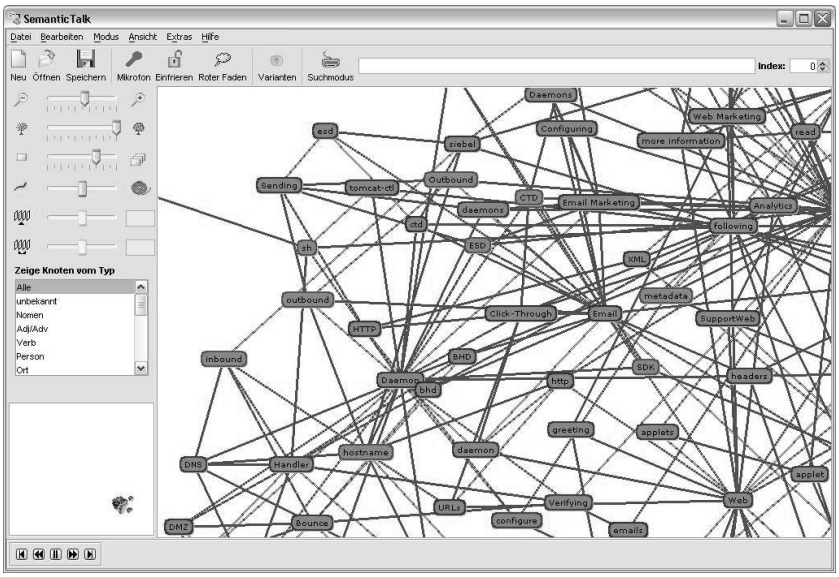


Abbildung 4: Benutzeroberfläche von SemanticTalk

4 Auswertung der semantisch aufbereiteten Inhalte

Die verschiedenen Möglichkeiten der Darstellungsanpassung erleichtern die Identifizierung anforderungsrelevanter Informationen. Insbesondere das attributspezifische Ein- und Ausblenden von Teilen des Graphen unterstützt vielfältige analytische Herangehensweisen. Da jeder Knoten des Graphen auf die entsprechenden Stellen in den Dokumentenbeständen verweist, kann auf spezifische Themen isoliert zugegriffen werden.

Abbildung 5 zeigt einen etwas größeren Ausschnitt des semantischen Netzes, das aus der CRM-Systembescheibung generiert wurde (vgl. Abbildung 4.). Die Visualisierung lässt mehrere Cluster erkennen: Links ein Cluster mit Begriffen, die sich der Konfiguration des E-Mail-Servers und dem Handling im Rahmen von E-Mail-Marketingkampagnen zuordnen lassen; in der Mitte ein Cluster mit Begriffen aus dem Online-Marketing, die häufig im Zusammenhang mit Newslettern und Kunden-Downloads auftreten; rechts unten ein Cluster zum Thema Benutzerschnittstelle und Nutzerinteraktion. Zwischen den beiden oberen Clustern fällt der Begriff „Web“ auf, von dem eine große Zahl Kanten in alle Richtungen abgeht. Solch sternförmig vernetzte Knoten finden sich an mehreren Stellen im Graphen – hierbei handelt es sich in der Regel um zentrale Begriffe eines Clusters oder Bindeglieder zwischen Clustern.

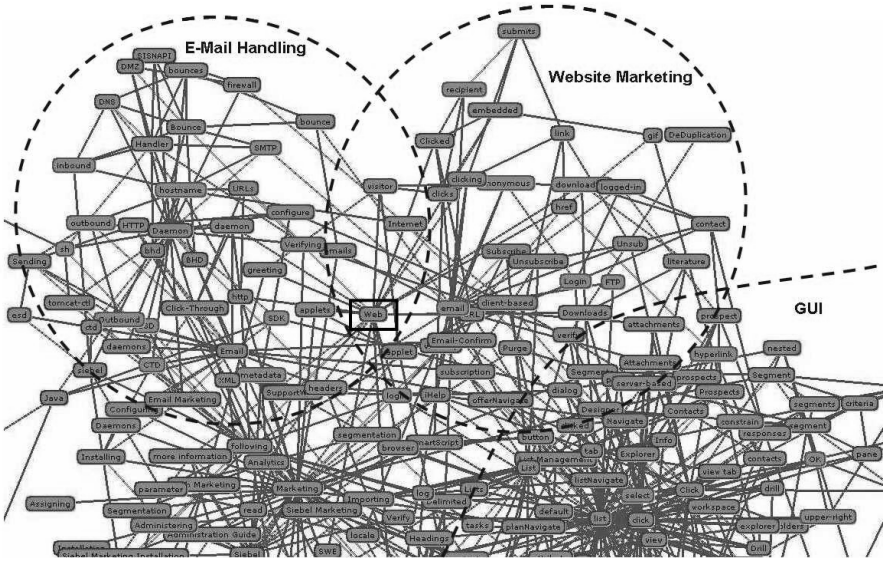


Abbildung 5: Ausschnitt eines semantischen Netzes einer CRM-Systembeschreibung

Die Visualisierung kann nützliche Anreize für domänenspezifische Kategorisierungen von Anforderungen liefern oder auf Begriffe aufmerksam machen, die eine Entsprechung in der Projektontologie (vgl. [LZ07, RL07]) bzw. dem Projektglossar haben sollten. Sofern geeignet, können bestimmte Teile des semantischen Netzes sogar direkt in die Projektontologie übernommen werden. Semantische Beschreibungen für die Begriffe lassen sich dann bestenfalls aus den referenzierten Stellen in den jeweiligen Dokumenten aufgreifen. Soll beispielsweise ein Re-Engineering im Zusammenhang mit dem genannten CRM-System durchgeführt werden, könnten Anforderungen unter anderem anhand der identifizierten Cluster „E-Mail Handling“, „Website Marketing“ und „GUI“ (vgl. Abbildung 5) klassifiziert werden. Fachbegriffe und Akronyme wie „Daemon“ oder „DMZ“ sollten in die Projektontologie bzw. das Projektglossar übernommen und semantisch eindeutig beschrieben werden.

Darüber hinaus kann die Visualisierung der semantischen Struktur kreative Prozesse und die Ideenfindung im Rahmen der Anforderungserhebung anregen. In anderem Zusammenhang haben wir mit dieser Form der Themenextraktion und -visualisierung semantische Kontexte für Gruppensitzungen erzeugt, die zu neuen Impulsen für den Diskussionsverlauf führten und die Gruppenkreativität anregten [Zi05].

5 Fazit und zukünftige Arbeiten

Wie in diesem Beitrag dargestellt wurde, kann die semantische Aufbereitung von schwach strukturierten, heterogenen Dokumentenbeständen zu einem Mehrwert für die Anforderungserhebung führen, der sich insbesondere in verringertem Auswertungsaufwand, verbessertem Überblick sowie themenbezogenem Zugriff auf die Inhalte manifestiert. Die Visualisierung der semantischen Struktur kann kreative Prozesse und die Ideenfindung anregen sowie die Identifizierung von domänenspezifischen Anforderungskategorien und Konzepten für die Projektontologie unterstützen. Anforderungsrelevante Teile des semantischen Netzes sollen in Zukunft unmittelbar in die kollaborative SoftWiki-Umgebung übernommen und dort erweitert und verfeinert werden können. Dies soll helfen, den häufig zu beobachtenden Bruch zwischen kreativen Vorstufen und formalen Modellierungsaktivitäten zu verringern. Da zu jedem Begriffsknoten die referenzierten Textstellen in den Dokumentenbeständen hinterlegt sind, bleibt eine hohe Traceability [RJ01] gewahrt: Es lässt sich leicht nachverfolgen, aus welchen Teilen der Dokumente Anforderungen hervorgegangen sind.

Zukünftige Arbeiten umfassen die Entwicklung gemeinsamer Schnittstellen mit der kollaborativen SoftWiki-Umgebung, um einen effizienten Transfer der semantischen Strukturen zu ermöglichen. Zum einen soll es in Zukunft möglich sein, auf komfortable Weise Teile des semantischen Netzes herauszulösen und in die kollaborative SoftWiki-Umgebung zu integrieren. Andersherum sollen auch in der kollaborativen Umgebung erzeugte RDF-Strukturen in Form von semantischen Netzen visualisiert und editiert werden können.

Danksagung

Wir bedanken uns bei Oliver Pape, Christian Räther und Sabine Köhler von der ISA Informationssysteme GmbH für ihren Input zum vorliegenden Beitrag.

Literaturverzeichnis

[Au06] Auer, S.; Riechert, T.; Fähnrich, K.-P.: SoftWiki – Agiles Requirements-Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder. In (Meißner, K.; Engelien, M.; Hrsg.): Virtuelle Organisation und Neue Medien 2006: Workshop GeNeMe2006. Gemeinschaften in Neuen Medien. TUDpress, Dresden, 2006.

[FS06] Feldman, R.; Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2006.

[Gr82] Griffith, R.L.: Three Principles of Representation for Semantic Networks. ACM Transactions on Database Systems, Vol. 7, Nr. 3, 1982; S. 417-442

[Ha07] Hagen, M.; Jungmann, B.; Lauenroth, K.: Ein Prozessmodell für ein agiles und wikibasiertes Requirements Engineering mit Unterstützung durch Semantic-Web-Technologien. In: Proceedings of 1st Conference on Social Semantic Web, LNI, Köllen-Verlag, Bonn, 2007.

[HQW06] Heyer, G.; Quasthoff, U.; Wittig, T.: Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse. W3L-Verlag, Herdecke, Bochum, 2006.

[HQW02] Heyer, G.; Quasthoff, U.; Wolff, C.: Automatic Analysis of Large Text Corpora - A Contribution to Structuring WEB Communities. In: Proceedings of the 2nd International Workshop on Innovative Internet Computing Systems (IICS), Springer-Verlag, Berlin, Heidelberg, 2002; S. 15-26

[KGV83] Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P.: Optimization by Simulated Annealing. Science, Vol. 220, Nr. 4598, 1983; S. 671-680

[Mc04] McBride, B.: RDF and its Vocabulary Description Language. In: Handbook on Ontologies, Springer-Verlag, Berlin, Heidelberg, 2004.

[LS99] Lassila, O.; Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 22 Februar 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222

[LZ07] Lohmann, S.; Ziegler, J.: Partizipationsformen und Entwicklung eines gemeinsamen Verständnisses bei der verteilten Anforderungserhebung. In: Proceedings of 1st Conference on Social Semantic Web, LNI, Köllen-Verlag, Bonn, 2007.

[RJ01] Ramesh B., Jarke M.: Towards Reference Models for Requirements Traceability. IEEE Transactions in Software Engineering, Vol. 27, Nr. 1, 2001; pp. 58-93

[RL07] Riechert, T.; Lohmann, S.: Mapping Cognitive Models to Social Spaces – Collaborative Development of Project Ontologies. In: Proceedings of 1st Conference on Social Semantic Web, LNI, Köllen-Verlag, Bonn, 2007.

[So91] Sowa, J.F. (Hrsg.): Principles of Semantic Networks: Explorations in the Representation of Knowledge, Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[ST05] Steyvers, M., Tenenbaum, J.B.: The Large-Scale Structure of Semantic Networks: Statistical Analyses and a Model of Semantic Growth. Cognitive Science, Vol. 29, Nr. 1, 2005; S. 41-78

[Zi04] Ziegler, J.; El Jerroudi, Z.; Böhm, C.; Beinhauer, W.; Busch, R.; Räther, C.: Automatische Themenextraktion aus gesprochener Sprache. In (Keil-Slawik, R.; Selke, H.; Szwillus, G., Hrsg.): Mensch & Computer 2004: Allgegenwärtige Interaktion. Oldenbourg Verlag, München, 2004; S. 281-290.

Partizipationsformen und Entwicklung eines gemeinsamen Verständnisses bei der verteilten Anforderungserhebung

Steffen Lohmann, Jürgen Ziegler

Universität Duisburg-Essen
Abt. Informatik und Angew. Kognitionswissenschaft
Lotharstrasse 65, 47057 Duisburg
{lohmann, ziegler}@interactivesystems.info

Abstract: Die webbasierte, verteilte Anforderungserhebung geht mit Herausforderungen einher, die neuartige Partizipations- und Interaktionsformen notwendig machen. Vor diesem Hintergrund präsentieren wir im vorliegenden Beitrag Lösungsansätze, die im Rahmen des SoftWiki-Projekts entwickelt werden. Zunächst erläutern und vergleichen wir die unterstützten Formen der Stakeholder-Partizipation. Anschließend thematisieren wir die semantische Anreicherung von Anforderungen. Hierbei stellen wir einen Ansatz vor, bei dem durch gemeinsame Verschlagwortung von Stakeholder-Beiträgen ein kollektiver Begriffsraum entsteht, der sich als Ausgangspunkt für die Klassifizierung von Anforderungen und die Entwicklung eines gemeinsamen Verständnisses vom zu entwickelnden Produkt eignet.

1 Einleitung

Ein zentrales Ziel des SoftWiki-Ansatzes ist es, möglichst viele Stakeholder in die Anforderungserhebung einzubeziehen [Au06]. Die Stakeholder sollen befähigt werden, sich per Webbrowser unmittelbar zu beteiligen. Dadurch entstehen verschiedenartige Herausforderungen an die zu realisierenden Interaktions- und Partizipationsformen. Zu Beginn der Anforderungserhebung besitzen die Stakeholder in aller Regel eine sehr heterogene Vorstellung vom zu entwickelnden Produkt [Ro01]. Zentrale Begriffe, deren Bedeutungen und Beziehungen zueinander werden unterschiedlich aufgefasst. Hier gilt es, innerhalb der Stakeholder ein gemeinsames Verständnis hinsichtlich des zu entwickelnden Produkts zu schaffen und auf dieser Basis die Beiträge der Stakeholder zu strukturieren.

Den übergeordneten Rahmen für die semantische Strukturierung von Anforderungen bildet die SoftWiki Ontology for Requirements Engineering (SWORE, vgl. [RLL07]). Sie stellt den ‚kleinsten gemeinsamen Nenner' für die Anforderungserhebung im SoftWiki-Ansatz dar: Letztlich müssen alle Anforderungen Instanzen von SWORE-Klassen sein. Wie in Abbildung 1 schematisch dargestellt, existiert in aller Regel zusätzlich noch eine große Anzahl weiterer, projekt- und domänenspezifischer Entitäten, auf die sich Beiträge von Stakeholdern beziehen und die in der SWORE durch das Konzept der Referenzpunkte berücksichtigt sind. Die Gesamtheit dieser Entitäten und ihrer Beziehungen

zueinander nennen wir die Projektontologie. Diese ist am ehesten vergleichbar mit einer Kombination aus (1) einem Glossar, das zentrale Begriffe des Projektkontextes beschreibt, (2) Relationen, die diese Begriffe miteinander verbinden, und (3) dem formal dokumentierten, gemeinsamen Produktverständnis aller Stakeholder. Die Projektontologie beinhaltet zwei Arten von Entitäten: Einerseits Entitäten, die nur im Kontext des Projekts eine sinnvolle Existenz erhalten, und andererseits Entitäten, die allgemeingültiges Wissen repräsentieren, das auch außerhalb des Projekts Bedeutung besitzt und eventuell sogar bereits im Web als Domänenwissen verfügbar ist.

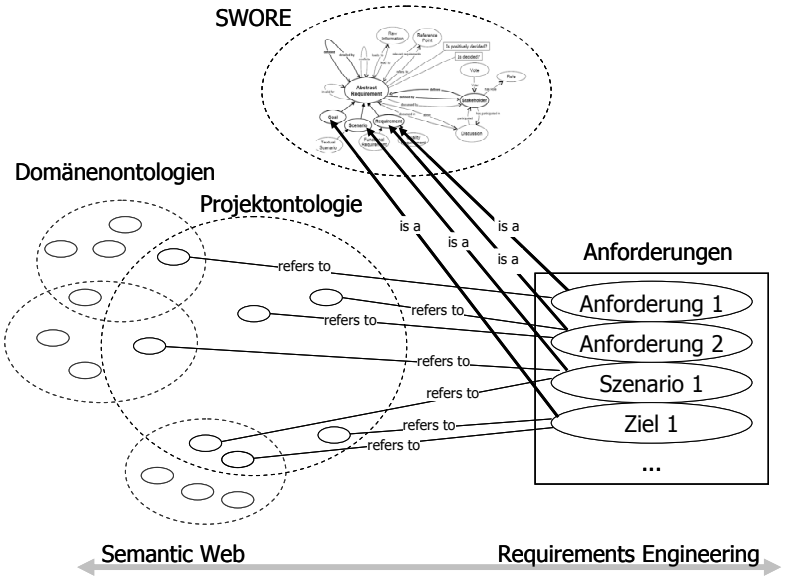


Abbildung 1: Anforderungen beziehen sich auf Konzepte der SWORE- und der Projektontologie

Die Erstellung von Ontologien ist eine Disziplin für sich, die einige Erfahrung verlangt [GFC04]. Um dennoch möglichst viele Stakeholder zu befähigen, sich direkt an der webbasierten Anforderungserhebung und der Entwicklung eines gemeinsamen Produktverständnisses zu beteiligen, werden im SoftWiki-Ansatz verschiedene Formen der Nutzerpartizipation und -interaktion entwickelt und erfolgreiche Konzepte aus dem Bereich Social Software aufgegriffen.

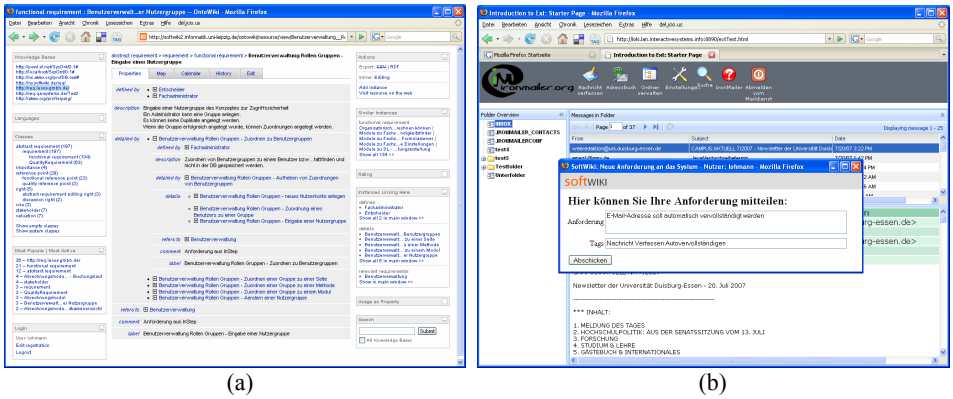
Im Folgenden werden zunächst die unterstützten Formen der Stakeholder-Partizipation erläutert und miteinander verglichen. Anschließend wird ein Ansatz vorgestellt, bei dem durch freie Verschlagwortung von Stakeholder-Beiträgen ein gemeinsamer Begriffsraum entsteht, der sich als Ausgangspunkt für die Klassifizierung von Anforderungen und die Entwicklung eines gemeinsamen Verständnisses vom zu entwickelnden Produkt eignet.

2 Partizipationsformen bei der verteilten Anforderungserhebung

Im SoftWiki-Projekt werden verschiedene Benutzerschnittstellen entwickelt, über die sich Stakeholder am Prozess der Anforderungserhebung beteiligen können. Die Wahl der geeigneten Benutzerschnittstelle ist stark vom jeweiligen Kontext abhängig. Sie wird insbesondere durch die Rolle des Stakeholders sowie seine Vorkenntnisse, Fähigkeiten und Motivationen determiniert, aber auch durch die bevorzugte Arbeitsumgebung und das Maß, in dem bereits eine Projektontologie existiert. Im Wesentlichen unterscheiden wir im SoftWiki-Ansatz zwei Organisationsformen, über die Stakeholder partizipieren können: die zentrale und die dezentrale Anforderungserhebung.

2.1 Zentrale Anforderungserhebung

Die zentrale Umgebung bietet dem Wiki-Konzept [LC01] folgend eine intuitive, leicht verständliche Benutzeroberfläche an, auf der sich Stakeholder ohne lange Einarbeitungszeit an der Anforderungserhebung beteiligen können (vgl. [Au06] und Abbildung 2a). Die Motivation zur Partizipation kann sowohl intrinsischer als auch extrinsischer Natur sein: Beispiele sind ein persönliches oder berufliches Interesse an der erfolgreichen Durchführung des Projekts, die Verantwortlichkeit qua Amt oder eine Steigerung der Reputation durch aktive Beteiligung. In jedem Fall verlangt diese Umgebung vom Stakeholder gewisses Engagement und die Bereitschaft, sich mit den Beiträgen anderer Stakeholder auseinander zu setzen. Der Nutzer wird bei der Interaktion mit der Umgebung kaum geführt, sondern muss selbst entscheiden, in welcher Weise er vorgeht. Im Gegenzug bietet die zentrale Plattform vielfältige Möglichkeiten, um auf den Prozess der Anforderungserhebung einzuwirken.



(a) (b)

Abbildung 2: (a) Zentrale und (b) dezentrale Anforderungserhebung

2.2 Dezentrale Anforderungserhebung

Auch wenn die strikte Befolgung der Wiki-Philosophie ‚making it easy to fix mistakes rather than making it hard to make them' zu einer hohen Beteiligungsbereitschaft bei-

tragen soll, wird man mit der zentralen Plattform einen großen Teil der Stakeholder nicht erreichen. Insbesondere Anwendern, die schnell und einfach ihre persönlichen Ideen und Anforderungen mitteilen wollen, ohne sich näher mit den kollaborativ erstellten Inhalten auseinanderzusetzen, sollten alternative, für diesen Zweck optimierte Nutzerschnittstellen zur Verfügung gestellt werden. Aus diesem Grund wird im Rahmen des SoftWiki-Projekts die wiki-basierten Oberfläche um eine dezentrale Variante der Anforderungserfassung erweitert, die sich auf einfache Weise in verschiedenste webbasierte Systeme integrieren lässt (vgl. Abbildung 2b). Mit diesem portablen Ansatz der Anforderungserfassung werden auch Anwendergruppen eingebunden, die mit der zentralen Wiki-Umgebung nicht zu erreichen sind.

Soll in der Anforderungserhebung beispielsweise Bezug auf eine bereits existierende Anwendung genommen werden, kann die dezentrale Umgebung in diese Anwendung integriert werden und Endanwender zur Eingabe von konstruktiven Verbesserungsvorschlägen auffordern. Die Endanwender können unmittelbar beitragen, ohne ihre eigentlichen Tätigkeiten unnötig lange zu unterbrechen. Sie müssen nicht erst in eine andere Umgebung wechseln, sich dort orientieren und überlegen, an welcher Stelle sie ihren Beitrag geeignet unterbringen. Der Anwender wird in seiner Umgebung abgeholt und aktiv zur Beteiligung aufgefordert.

Während die zentrale Umgebung dem Nutzer großen Handlungsspielraum einräumt und ihm eine Vielzahl an Partizipationsmöglichkeiten anbietet, engt die dezentrale Umgebung seine Freiheitsgrade zugunsten unmittelbarer Verständlichkeit und stärkerer Lenkung bewusst ein. Tabelle 1 stellt wesentliche Merkmale der zentralen und dezentralen Anforderungserhebung einander gegenüber.

	Zentrale Anforderungserhebung	**Dezentrale Anforderungserhebung**
Applikationstyp	eigenständige Anwendung	integrierte Anwendung
Kooperationsgrad	hoch	gering
Nutzerführung	gering	hoch
Redundanz in den Beiträgen	gering	hoch
Aktionsform	Beiträge werden eingestellt (Push)	Beiträge werden abgeholt (Pull)
Nutzergruppe	v.a. Verantwortliche	v.a. Endanwender
Aufwand	mittel bis hoch	gering bis mittel
Freiheitsgrade	viele	wenige
Einfachheit der Bedienung	leicht erlernbar	unmittelbar verständlich

Tabelle 1: Gegenüberstellung wesentlicher Merkmale der zentralen und der dezentralen Anforderungserhebung

3 Zuweisung von Referenzpunkten

Wie eingangs erläutert, sollen alle Stakeholder-Beiträge letztlich semantisch eindeutigen Referenzpunkten zugewiesen werden, die dabei helfen, die Beiträge zu strukturieren und semantisch anzureichern. Allgemein lassen sich drei Arten der Zuweisung von Referenzpunkten unterscheiden:

1. Der Verweis auf einen bereits bestehenden Referenzpunkt
2. Die Erstellung eines neuen Referenzpunktes
3. Die automatische Ermittlung eines passenden Referenzpunktes

3.1 Zuweisung von Ontologieinstanzen

Eine Form der Zuweisung von Referenzpunkten zu Stakeholder-Beiträgen besteht darin, die Beiträge direkt mit Instanzen der Projektontologie zu verknüpfen. Existiert keine passende Instanz, erstellt der Nutzer eine neue. Hierfür bietet die zentrale SoftWiki-Umgebung einfache Erstellmöglichkeiten nach dem Wiki-Ansatz, die sich leicht erlernen und anwenden lässt. Trotz des Wiki-Ansatzes verlangt die Zuweisung von Instanzen gewissen Aufwand: Der Nutzer muss sich die Projektontologie anschauen und entscheiden, mit welcher Instanz er seinen Beitrag verbindet, ob die Instanz den Beitrag angemessen reflektiert oder ob nicht vielmehr eine neue Instanz zu erstellen bzw. die vorhandene anzupassen ist. Um die Zuweisung und Erstellung von Ontologieinstanzen zu unterstützen, sollen zukünftig durch Text Mining-Verfahren [HQW06] automatisch Vorschläge für möglicherweise passende Instanzen generiert werden.

3.2 Zuweisung von Schlagworten

Die alternative Möglichkeit der Zuweisung sieht eine gemeinsame Verschlagwortung der Beiträge vor. Hierbei versehen die Stakeholder ihre eigenen Beiträge oder Beiträge anderer mit einer beliebigen Anzahl frei gewählter Schlagworte. Auch diese Art der Zuweisung soll durch Text Mining-Verfahren unterstützt werden, indem zusätzlich aus dem Beitrag des Stakeholders automatisch Schlüsselbegriffe extrahiert werden. Das Resultat dieser kombinierten manuellen und automatischen Indexierung wird in einem Tupel folgender Form festgehalten:

```
(Nutzereingabe, {Nutzerschlagworte}, {Extrahierte Schlüs-
selbegriffe}, Nutzer-ID)
```

Bezogen auf ein Beispielszenario zur Anforderungserhebung für eine E-Mail-Anwendung könnte dieses Tupel beispielsweise wie folgt ausgefüllt sein:

```
(„E-Mail-Adresse soll automatisch vervollständigt werden“,
{Nachricht, Verfassen, Autovervollständigen}, {E-Mail, Ad-
resse, automatisch}, 102)
```

3.3 Kooperativer Rahmen

Im SoftWiki-Projekt verfolgen wir die Realisierung beider Zuweisungsarten sowohl für die zentrale als auch für die dezentrale Form der Anforderungserhebung, allerdings mit der bewussten Einschränkung, dass in der dezentralen Umgebung die Bearbeitung der Projektontologie nicht möglich ist. In einem Großteil der Situationen wird die freie Verschlagwortung die bevorzugte Form der Zuweisung sein: Auf einfache Weise können Beiträge unter Verwendung des eigenen Vokabulars mit beliebigen Schlagworten versehen werden [Si05]. Deshalb soll auch in der zentralen, kollaborativen Umgebung zukünftig neben der Zuweisung von Ontologieinstanzen die Auszeichnung von Beiträgen mit Schlagworten möglich sein

Über die zentrale Umgebung sind beide Vorgehen in einen kooperativen Rahmen eingebunden: Die beigetragenen Inhalte und zugewiesenen Referenzpunkte der dezentralen Anforderungserhebung werden ebenfalls in der zentralen Umgebung verwaltet und können von allen Stakeholdern eingesehen werden. Die Stakeholder können den Beiträgen neue Referenzpunkte zuweisen oder bestehende Referenzpunkte anpassen. Ontologieinstanzen und Schlagworte werden hierbei unterschiedlich behandelt: Während Ontologieinstanzen ausschließlich global verändert werden können, sind Schlagworte personengebunden.

Mit dem beschriebenen Ansatz werden zwei verschiedene Vorgehensweisen bedient:

1. Die unmittelbare, kollaborative Erstellung einer gemeinsamen Projektontologie. Dieses Vorgehen fällt in den Bereich Knowledge Engineering und verlangt den beteiligten Stakeholdern eine gewisse Einarbeitung in die Umgebung ab. Durch die Umsetzung des Wiki-Konzepts ist der damit verbundene Aufwand jedoch wesentlich reduziert.

2. Die Erstellung eines gemeinsamen Begriffraums durch die freie Verschlagwortung der Beiträge. Dieses Vorgehen erfreut sich derzeit insbesondere im Webumfeld unter Bezeichnungen wie *Collaborative* oder *Social Tagging* großer Beliebtheit. Die Einfachheit dieses Ansatzes geht mit einer hohen Nutzerakzeptanz und geringen Partizipationsschwelle einher.

Letztere Vorgehensweise hat jedoch den Nachteil, dass der resultierende Begriffsraum nur eine geringe semantische Formalität aufweist: Als Schlagworte können beliebige Bezeichner gewählt werden; die Bedeutungen der Bezeichner sowie Beziehungen zwischen diesen werden nicht explizit angegeben. Dennoch kann der Begriffsraum die Herausbildung eines gemeinsamen Verständnisses fördern und eine inspirierende Quelle für weitere Anforderungen und Ideen darstellen. Darüber hinaus kann der Begriffsraum einen wertvollen Ausgangspunkt für die Erstellung der Projektontologie bilden und das ‚Kaltstart'-Problem verringern, das sich häufig zu Beginn der Ontologiemodellierung ergibt [GFC04].

5 Entwicklung eines gemeinsamen Verständnisses

Bei der Verschlagwortung der Beiträge werden die Stakeholder Begriffe verwenden, die ihrer persönlichen Vorstellung vom zu entwickelnden Produkt entsprechen. Ihr mentales Produktmodell spiegelt sich damit zu einem gewissen Grad in den Schlagworten wider. Der sukzessiv entstehende, gemeinsame Begriffsraum projiziert folglich in einer vereinfachten Form das kollektive Verständnis vom zu entwickelnden Produkt. Zugleich werden die Stakeholder-Beiträge anhand dieses Begriffsraums strukturiert. Die Begriffe bilden thematische Kategorien, über die die Beiträge exploriert und ausgewertet werden können.

Um einen möglichst homogenen Begriffsraum zu erzeugen, sollten die verwendeten Schlagworte vor ihrer Aggregation normalisiert werden. Hierzu können verschiedene automatische Verfahren eingesetzt werden: Von der Konvertierung in Kleinbuchstaben und dem Entfernen von Satz- und Sonderzeichen über eine Grundformreduktion bis hin zur Korrektur von Tipp- und Rechtschreibfehlern [LZ07]. Durch die Etablierung von Konventionen bei der Verschlagwortung kann eine zusätzliche Homogenität erzielt werden. Konventionen bilden sich jedoch nicht in allen Kontexten heraus, so dass sich in vielen Fällen eine umfangreiche Normalisierung als zweckdienlich herausstellen wird.

5.1 Multiperspektivische Visualisierung

Um die Identifizierung von Gemeinsamkeiten und Unterschieden zwischen den Sichten verschiedener Stakeholder zu unterstützen, verfolgen wir einen Ansatz, der die Exploration von Begriffsräumen aus unterschiedlichen Perspektiven ermöglicht. Hierbei unterscheiden wir folgende Sichten:

- Die *Persönliche Sicht* umfasst alle Begriffe, die ein einzelner Stakeholder Beiträgen zugeordnet hat.
- Die *Gesamtsicht* gibt die Gesamtheit an Begriffen wieder, die von allen Stakeholdern den Beiträgen zugeordnet wurden.
- Die *Extrahierte Sicht* zeigt die Gesamtheit an Begriffen, die automatisch aus allen Beiträgen der Stakeholder extrahiert wurden.

Abbildung 3 zeigt eine beispielhafte Implementierung dieser verschiedenen Sichten in Zusammenhang mit einem Semantic Web-Projekt. Für die Darstellung wurde eine Begriffswolken (Tag Cloud)-Visualisierung gewählt – eine zweidimensionale, gewichtete Liste, in der alle bzw. die am häufigsten verwendeten Begriffe alphabetisch sortiert aufgeführt sind. Die Visualisierung mittels einer Begriffswolke ermöglicht einen schnellen Überblick über zentrale Begriffe und deren relative Verwendungshäufigkeit [Ri07]. In der abgebildeten Umsetzung werden die verschiedenen Sichten einander direkt gegenübergestellt und Unterschiede farblich hervorgehoben, wodurch eine bessere Vergleichbarkeit erzielt wird.

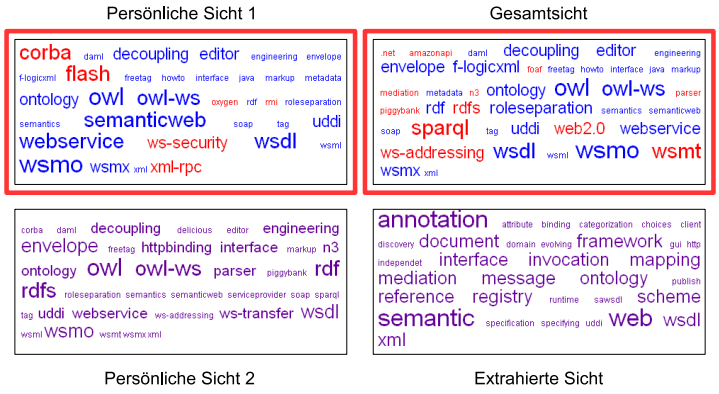


Abbildung 3: Verschiedene Sichten auf einen gemeinsamen Begriffsraum

5.2 Semantische Anreicherung

Der gemeinsame Begriffsraum spiegelt bereits einen Teil des Projektverständnisses der Stakeholder wider und eignet sich damit als Ausgangspunkt für die Erstellung der Projektontologie bzw. die Erweiterung einer bereits bestehenden Projektontologie. Um diesen Prozess zu erleichtern, wollen wir zukünftig signifikante Kookkurrenzen zwischen den Schlagworten ermitteln lassen: Werden zwei Begriffe überdurchschnittlich häufig gemeinsam bei der Verschlagwortung verwendet, wird ein thematischer Zusammenhang zwischen diesen Begriffen angenommen [MC07]. Um die ermittelten Beziehungen darzustellen und zu bearbeiten, kann die zentrale Wiki-Umgebung verwendet werden. Darüber hinaus soll die Darstellung und Bearbeitung in einer Visualisierung als Semantischen Netz möglich sein (vgl. [Cy07]) und die Begriffswolken-Visualisierung so erweitert werden, dass durch entsprechende räumliche Anordnung die semantische Verwandtschaft zwischen Begriffen näherungsweise dargestellt wird [HH07]. Eine automatische Typisierung der Begriffsrelationen ist im Fall von Verschlagwortungen jedoch kaum möglich [LZ07].

Außerdem ist geplant, die Begriffe semi-automatisch um weitere Semantik anzureichern. Zur Unterstützung dieses Prozesses werden einerseits im Projektkontext existierende Dokumentenbestände herangezogen und semantisch aufbereitet [Cy07] und andererseits auf im Web verfügbares Domänenwissen zugegriffen (vgl. Abbildung 1). Ansätze wie DBpedia [AL07] verfolgen das Ziel, dieses implizit vorhandene Wissen zu extrahieren und strukturiert zur Verfügung zu stellen.

6 Fazit

Die Bereitstellung einer zentralen, webbasierten Umgebung, über die sich Stakeholder an der verteilten Anforderungserhebung beteiligen können, wird alleine nicht ausreichen, um wirklich hohe Partizipation zu erzielen. Die Stakeholder müssen letztlich an dem Punkt abgeholt werden, an dem sie bereit sind, sich zu beteiligen. Vor diesem Hintergrund wurden die zentrale und dezentrale Anforderungserhebung sowie verschiedene Varianten der semantischen Anreicherung von Anforderungen thematisiert. Es wurde verdeutlicht, dass sich möglichst viele und unterschiedliche Stakeholdergruppen nur über die Bereitstellung verschiedener Partizipations- und Interaktionsformen einbeziehen lassen, die jedoch in einen gemeinsamen kooperativen Rahmen eingebunden sein sollten.

Literaturverzeichnis

[AL07] Auer, S.; Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. In (Franconi, E.; Kifer, M.; May, W.): Proceedings of the 4th European Semantic Web Conference, LNCS 4519, Springer, 2007; 503-517.

[Au06] Auer, S.; Riechert, T.; Fährich, K.-P.: SoftWiki – Agiles Requirements-Engineering für Softwareprojekte mit einer großen Anzahl verteilter Stakeholder. In (Meißner, K.; Engelien, M.; Hrsg.) Virtuelle Organisation und Neue Medien 2006: Workshop GeNeMe2006. Gemeinschaften in Neuen Medien. TUDpress, 2006.

[Cy07] Cyriaks, H.; Lohmann, S.; Velioglu, V.; Ziegler, J.: Semantische Aufbereitung von Dokumentenbeständen zur Gewinnung anforderungsrelevanter Informationen. In: Proceedings of 1st Conference on Social Semantic Web, LNI, Köllen-Verlag, Bonn, 2007.

[GFC04] Gómez-Pérez, A.; Fernández-López, M.; Corcho-Garcia, O.: Ontological Engineering. Springer, 2004.

[HH07] Hassan-Montero, Y.; Herrero-Solana, V.: Improving Tag-Clouds as Visual Information Retrieval Interfaces. In: Current Research in Information Sciences and Technologies. Multidisciplinary approaches to global information systems, 2006.

[HQW06] Heyer, G.; Quasthoff, U.; Wittig, T.: Text Mining: Wissensrohstoff Text – Konzepte, Algorithmen, Ergebnisse. W3L-Verlag, Herdecke, Bochum, 2006.

[LC01] Leuf, B.; Cunningham, W.: The Wiki Way: Collaboration and Sharing on the Internet. Addison-Wesley Professional, 2001.

[LZ07] Lohmann, S.; Ziegler, J.: Bringing Semantics into Folksonomies – Semantische Analyse nutzergenierter Indexierungen. In: Informatik 2007 – Informatik trifft Logistik!, Köllen-Verlag, 2007.

[MC07] Michlmayr, E.; Cayzer, S.: Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. In: Proc. of WWW 2007 Workshop on Tagging and Metadata for Social Information Organization, 2007.

[Ri07] Rivadeneira, A.W.; Gruen, D.M.; Muller, M.J.; Millen, D.R.: Getting our head in the clouds: Toward evaluation studies of tagclouds. In: Proceedings of the SIGCHI conference on Human factors in computing systems, 2007, S.: 995-998.

[RLL07] SWORE – SoftWiki Ontologie für Requirements Engineering. In: Proceedings of 1st Conference on Social Semantic Web, LNI, Köllen-Verlag, Bonn, 2007.

[Ro01] Robertson, S.: Are We Afraid of the Dark? IEEE Software, Vol. 18, Nr. 4, 2001; S. 12-15.

[Si05] Sinha, R.: A Cognitive Analysis of Tagging, http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html, 2005 (Stand: 01.06.07).

Galaxy: IBM Ontological Network Miner

John Judge, Mikhail Sogrin, Alexander Troussov

{johnjudge, sogrimik, atrousso}@ie.ibm.com

Abstract: Many applications of the semantic web and Web 2.0 aim to empower the knowledge worker. These applications however, do not allow the user to combine all of his/her social and semantic information into a single resource which allows data to be processed, managed and enhanced automatically. In our demo we will present a number of demo applications based on Galaxy, IBM's ontological network miner, which was designed to work with such resources to enhance the capabilities of a number of applications in social semantic computing. Galaxy is a highly efficient, scalable system which can be easily built into an application and can be optimised to suit a user's preferences or to take into account the needs of a particular task or application.

1 Introduction

Currently the semantic web relies on semantic annotations which, for the most part, are done manually by humans. Working in the EU 6^{th} framework project Nepomuk [Nepo] we in IBM Dublin have developed a tool which can be useful in the automation of metadata creation. Our ontological network miner, Galaxy, is a generic tool which performs elements of soft clustering on semantic networks such as company organisation trees, social networks and community diagrams or any other collection of data which can be represented by a graph network.

We perform automatic ontology-based conceptual tagging and find central concepts of a text with respect to the given lexico-semantic resource (ontology). For example, a text which mentions Mulhuddart, Lansdowne, Clontarf is probably about Dublin/Ireland/Europe/Earth. This fact can be inferred (assuming some geographical information exists in our semantic resource) from geographical relations like `Mulhuddart is-part-of Dublin`. Galaxy resolves any ambiguities on the fly based on the ontological knowledge from the corresponding semantic resource and uses the results of disambiguation in determining the results.

This kind of processing can be leveraged for numerous tasks including metadata generation, related item recommendation, community detection, and expert location. We have designed our application in such a way that it is highly configurable to make it adaptable to numerous tasks in social semantic computing.

The remaining sections of this paper are structured as follows: Section 3 describes our network mining algorithm and gives some performance statistics. Section 4 discusses some of the many applications which our algorithm could be used for. Finally Section 5

outlines some future directions for our work.

2 Motivation

[Tof90] observes that knowledge workers in the age of knowledge economies and knowledge societies need to have available to them a system which can be used to create, process, enhance and manage their knowledge and information. Recent advances in social computing and social semantic desktop applications are making such a system possible. However, many of the resources available for these tasks need significant manual intervention before they are useful to the knowledge worker. We have created a highly scalable and efficient ontology mining algorithm which can be used for a variety of tasks in social semantic computing and which can be developed into an application which can suggest links between resources to remove the need for manual intervention and which can be adapted to a user's preferences or for individual tasks.

The Nepomuk project aims to empower knowledge workers to better exploit their personal information space and to maintain fruitful communication and exchange within social networks irrespective of organizational boundaries. In the context of Nepomuk we are working with our partners to develop a comprehensive solution which extends the personal desktop to create a collaboration environment which supports both personal information creation, processing and management, and the sharing and exchange of information across social and organizational relations. This solution is called the Social Semantic Desktop (SSD).

The SSD is built upon the idea of a Personal Information Management Ontology (PIMO) which is a unified model of social and semantic data. The PIMO is neither a fixed nor a hierarchical entity, it grows and changes as the user creates new data, uses existing data and changes his/her social interactions. Because of the organic and dynamic nature of the PIMO an efficient, scalable method of mining information from the ontology and of inferring new data based on the topology is required to exploit this data fully.

Much of the existing network mining technology is lacking in this regard, often they rely on a rigid or hierarchical ontology structure or they suffer badly in terms of complexity on large datasets. For example [AHSS04] presents an accurate scalable algorithm which assigns a geographical focus to a texts based on mentions of places in the text. However their method requires that the underlying datasource is structured hierarchically and it is confined to just one domain of application. Galaxy is a significant improvement on these type of miners, not only is it efficient and scalable but because it makes no assumption about graph topology it can work on an ontology of any complexity.

3 Description

Galaxy is based on the spread of activation technique used in semantic networks. It is very flexible and customisable which allows it to be tailored for a range of applications. The spread of activation is used to find a focal node, or nodes, in the network based on the parameters and constraints given. Galaxy resolves any lexico-semantic ambiguities on the fly based on the ontological knowledge from the corresponding resource and uses the results of disambiguation in determining the focus.

Galaxy does not perform clustering in the traditional sense ("hard clustering"), where a graph or network is partitioned according to various clustering measures. Instead Galaxy performs a "fuzzy clustering" analysis, dealing with a (changing) sub-graph based around a set of nodes within the graph provided to it, and finds a focus (or foci) relative to those nodes and dependent upon the graph topology and the user's constraints on how propagation around the graph can happen. The focus found by Galaxy is similar to finding a central node or concept for the given sub-graph. However, dependent on the starting nodes, the constraints of the specific application and graph topology, multiple foci or no focus may be returned. In this way it does not return a central concept or focus unless one can be found which is close to the starting nodes.

Testing Galaxy as a stand alone entity is somewhat difficult. Standard metrics like precision and recall are difficult to apply without a particular task in mind and a specific test set and lexico-semantic resource for that task. Galaxy would also have to be incorporated into a "driver application" which would perform the particular task. This means that as yet we are unable to supply qualitative results on Galaxy's performance.

We have performed scalability tests on Galaxy to test how well the algorithm copes with different amounts of nodes activated in the initial query. The tests were carried out on a network of over 170,000 nodes, up to 100 initial nodes were chosen at random to activate and the time taken for each query length were averaged over 10 runs.

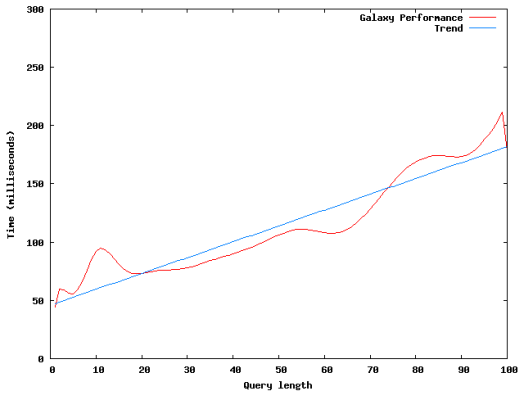


Figure 1: Graph of time versus query length performance for Galaxy

The graph in Figure 1 shows that the time taken to execute a query increases as the number of activated nodes in the query is increased. The overall trend in the time increase is roughly linear. The time taken to execute individual queries is remarkably small for such a large network, and even when the longest queries tested can execute in the region of 0.2 of a second.

4 Applications and the Demo

Galaxy can be used to add value to a range of different applications. We have developed a prototype application "workbench" which allows us to demonstrate a number of tasks useful for IBM's new enterprise corporate social software solution Lotus Connections [LSS]. We will demonstrate ways in which Galaxy can tie together resources from social software at IBM for a number of tasks including metadata generation, tag recommendation, community detection and expertise location.

Our demo will feature live demonstrations of these tasks and more applications as we discover them and refine our demonstration workbench. We will also demonstrate a composite application built on Lotus Notes 8 called Smart Assistant, which uses text analytics and Galaxy to organise incoming email and provide contextual information based on mail content and real world knowledge in the form of a semantic ontology.

5 Future Directions

As yet Galaxy is in the early stages of being deployed into real world situations and applications. We are working closely with our partners in Nepomuk, Digital Enterprises Research Institute Galway, and Trinity College Dublin to explore new areas of application and to build solutions based on Galaxy. This is an ongoing challenge and many new areas of research are open to us including using Galaxy as an application development component instead of just a monolithic ontology mining algorithm.

References

[AHSS04] Einat Amitay, Nadav Har'El, Ron Sivan, and Aya Soffer. Web-a-where: Geotagging Web Content. In *SIGIR*, pages 273–280, 2004.

[LSS] Lotus Social Software http://www.ibm.com/lotus/connections

[Nepo] Nepomuk - The Social Semantic Desktop http://nepomuk.semanticdesktop.org

[Tof90] Alvin Toffler. *Power Shift, Knowledge, Wealth and Violence at the Edge of the 21st Century*. Bantam Books, 1990.

IMAGENOTION - Collaborative Semantic Annotation of Images and Image Parts and Work Integrated Creation of Ontologies*

Andreas Walter, awalter@fzi.de
Gabor Nagypal, nagypal@disy.net

Abstract:
In this paper, we present the ImageNotion tool that allows for the semantic annotation of images and image parts together with the maturing of ontologies in a work integrated environment. The tool uses our ImageNotion methodology for ontology development.

Keywords Collaborative ontology engineering, image annotation, social software

1 Introduction

Currently, metadata about the content of images is largely based on the unstructured and non semantic tagging paradigm. E.g., also Flickr [Fli07] and Riya [Riy07], two popular systems for collaborative annotation of images, use this paradigm. Our current work aims to provide semantic search for image contents and also make easy navigation among images by simply clicking on their parts possible. This requires using domain specific ontologies for the annotation of images and image parts.

While tagging systems are user-friendly, ontology formalisms and development tools are too complicated for most users [FLGP02]. This fact normally leads to a seperation of the ontology engineering process from the usage of ontology for the semantic annotation of resources. When the content of an image repository rapidly changes — and this is the case for most image repositories that are created collaboratively — this separation usually results in missing or obsolete concepts in the ontology. I.e., an adequate and user-friendly annotation of images is not possible any more. Moreover, using separated ontology editors and image annotation tools raises the need for continuous ontology import/export between them. This makes the whole process cumbersome, slow and expensive even for experienced ontology engineers and domain experts.

We identified three challenges that must be solved to change this situation. First, the ontology development process should be simple enough that even the average user without much ontology experience can contribute to the creation of a meaningful ontology.

*This work was co-funded by the European Commission within the project IMAGINATION.

Second, the process of ontology development should be integrated into the process of semantic annotation. This work-integrated creation of the ontology eliminates the for the expensive communication among knowledge engineers and annotators (domain experts), and the need for importing/exporting ontologies. Work-integration allows the creation or adjustment of new ontology concepts exactly then, when the need for them arises during annotation: this makes ontology development well-motivated and intuitive. Finally, the ontology development process should also be collaborative so that users may profit from the work already done by their fellow users.

In this paper, we introduce our ImageNotion tool and methodology that allows for the collaborative, work-integrated development of ontologies and the semantic annotation of images and their parts. Thus, ImageNotion addresses all of the introduced challenges.

2 Related Work

The ideal solution for the collaborative, work-integrated development of ontologies would be a browser-based tool that is easy to use and follows an approach that allows both the development of ontologies and the semantic annotation of images (and their parts) in one integrated framework. Currently, we are not aware of any work that achieves this goal. Therefore we can only review works that address partial fields of our research.

Riya [Riy07] allows for a collaborative, browser based annotation of images and image parts with tags in a work-integrated environment. However, semantic annotation of images using ontologies is not possible. *Photostuff* [HWGS+06] is a stand-alone application that supports the semantic annotation of images with imported ontologies. Ontology development and collaborative work is not possible with this tool. Protégé [Pro07] is one of the most popular tools for ontology development. With an extension (Collaborative Protégé [TN07]), it also allows collaborative work. The problem of this tool is that it is difficult to use for non ontology experts. Semantic Wikis [VKV+06] are browser based and easy to use, but they are not suitable for the annotation of images.

A common drawback of all these tools is that a separate tool is required for semantic image annotation and therefore ontology development is no more work integrated. SOBOLEO [Bra07] is intended for non-ontology-experts, too, and allows the work-integrated maturing of ontologies. However, it is intended for the annotation of web pages and not for the annotation of images and especially image parts.

3 The ImageNotion Methodology

We tried to explain to our users (experienced in the area of tag based image annotation and thesauri) how to use the tools Protégé for the ontology development and PhotoStuff for the semantic image annotation. By using these tools, i.e., by separating the ontology building and semantic annotation processes, the semantic annotation of images was nearly

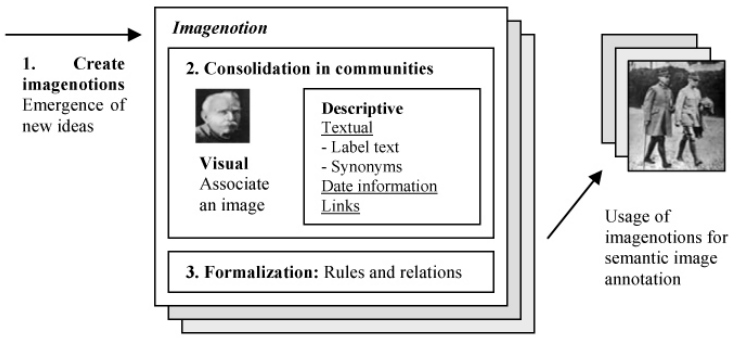


Figure 1: The ImageNotion methodology

impossible. The barriers were too high because of complexity tools, the complexity and formality of ontologies, and because of the cumbersome workflow for extending the ontology. This motivated our research for an approach more intuitive and more understandable for our users. The result is a methodology and a tool called ImageNotion.

Ontologies in the ImageNotion methodology consist of *imagenotions*. An imagenotion (formed from the words image and notion) graphically represents a semantic notion through an image. Our motivation was the ancient observation that "a picture worth a thousand words". Furthermore, similarly to many existing ontology formalisms, it is possible to associate descriptive information with an imagenotion. A part of the descriptive information is textual: labels in different languages (such as English or German). For each language, one of these synonymous labels is selected as the main label of the imagenotion. Other labels are termed synonyms. Additionally, date information (exact date or time interval) attached to the imagenotion allows for temporal queries on images. Further, it is possible to add links to web pages for an imagenotion. Links provide additional background information for the users of the ontology. On the one hand, this makes imagenotions easier to comprehend. On the other hand, it is easier for the users to extend the description of the imagenotion based on this background information, i.e., to mature the imagenotion. In addition to descriptive information, relations among imagenotions are also possible. Currently we support hierarchical relations (broader and narrower imagenotions). All other relations are termed "unnamed relations".

3.1 The Phases of the ImageNotion Methodology

The aim of the ImageNotion methodology (see Fig. 1) is to guide the process of visually creating an ontology that contains imagenotions and relations among them. The main steps of this methodology are based on the ontology maturing process model that we have described in [BNS+06]. Step 1 is the creation of new imagenotions, step 2 is the consolidation of imagenotions in communities and step 3 is the formalization of imagenotions with rules and relations. Imagenotions from each maturing grade may be used for semantic image annotation.

Figure 2: Creation of new imagenotions

3.2 Social Ascpects of ImageNotion

The usual separation of ontology development and ontology usage leads to problems in practice, such as having outdated elements in the ontology ([Hep07]). Ontology maturing, i.e., the development of ontologies, is a social and collaborative process. This process should be supported by a user-friendly methodology that allows the needed collaboration. In ImageNotion, users can modify the underlying ontology[1] of a semantic application themselves and at once when the need arises. They also see the changes that are made by their fellow users. This approach, motivated by constructivist views on learning (see also [AMR06]), allows a community the creation of required ontologies that fit their needs as well as possible.

4 The ImageNotion Tool

We will now describe how the ImageNotion methodology is implemented in our tool. A demo version of the tool is accessible at `www.imagenotion.com/demo`.

Creating and Editing Imagenotions: Images with new ontology elements require the creation of new imagenotions. In Fig. 2, a user has new images of "Joseph Joffre" (french general in World War I). Since the ontology contains so far no concept (or instance, respectively) for Joffre, a new imagenotion is required that allows for the annotation of this new image. The user chooses one image in the archive showing Joffre and drags this image to the area that allows the creation of new imagenotions (see Fig. 2). Now she can enter a label in her preferred language and the new imagenotion is created. To add relations, the tool first allows searching for existing imagenotions. Then, it is possible to add these imagenotions as relations or to create a new imagenotion and relate it with the current one. E.g., in the example of "Joffre", relations to "France" or to "World War I" may be added (see Fig. 3).

[1] add new ontology elements or modify existing ones

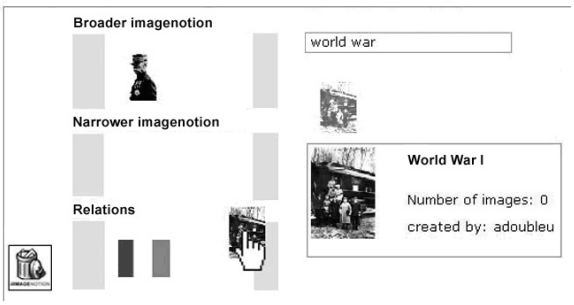


Figure 3: Adding relations between imagenotions

Annotation of Images and Image Parts: The annotation of images is possible with every available imagenotion. The tool allows searching for imagenotions and bookmarking the preferred ones as "my imagenotions". With drag and drop, a user can annotate images or image parts very easily. For image parts, the user can also specify the correct position of the image annotation box (see Fig. 4).

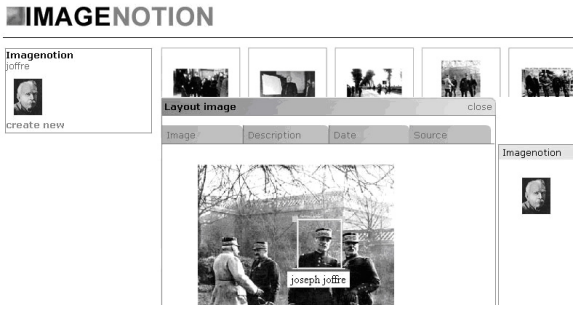


Figure 4: Semantic annotation of image parts

5 Conclusion and Future Work

The ImageNotion tool allows for the creation and maturing of ontologies for the semantic annotation of images and image parts. It is accessible via a standard web-browser and allows collaborative work in a work integrated environment. Our next work steps include implementing exporting possibilities for the the created ontologies, further elaborate on the issue of relations among imagenotions, and evaluating the system with experienced users in the area of image annotation within the EU project IMAGINATION[2]. In this project, we implement an expert-based version of the system, where only experts may annotate the images. Our vision is, however, the creation of a Flickr-like collaborative environment where imagenotions are used for the semantic search and annotation of image contents.

[2] `http://www.imagination-project.org`

References

[AMR06] Heidrum Allert, Hannu Markannen, and Christoph Richter. Rethinking the Use of Ontologies in Learning. In Martin Memmel and Daniel Burgos, editors, *Proceedings of the 2nd International Workshop on Learner-Oriented Knowledge Management and KM-Oriented Learning (LOKMOL 06), in conjunction with the First European Conference on Technology-Enhanced Learning (ECTEL 06)*, pages 115–125, October 2006.

[BNS+06] Braun, Nagypal, Schmidt, Walter, and Zacharias. Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering. In *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge , 16th International World Wide Web Conference (WWW 2007), Canada*, May 2007 2006.

[Bra07] Zacharias Braun. SOBOLEO - Social Bookmarking and Lightweight Ontology Engineering. In *Workshop on Social and Collaborative Construction of Structured Knowledge, 16th International World Wide Web Conference (WWW 2007), Canada*, 2007.

[FLGP02] Mariano Fernández-López and Asun Gómez-Pérez. A survey on methodologies for developing, maintaining, integrating, evaluating and reengineering ontologies. Deliverable 1.4, EU IST Project IST-2000-29243 OntoWeb, 2002.

[Fli07] Flickr. Welcome to Flickr - Photo Sharing. http://www.flickr.com/, 2007. (accessed 2007-05-31).

[Hep07] Martin Hepp. Possible Ontologies: How Reality Constraints Building Relevant Ontologies. *IEEE Internet Computing*, 11(1):90–96, January/February 2007.

[HWGS+06] Halaschek-Wiener, Golbeck, Schain, Grove, Parsia, and Hendler. Annotation and provenance tracking in semantic web photo libraries. In *International provenance and annotation workshop*, 2006. (accessed 2007-05-31).

[Pro07] Protégé. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/, 2007. (accessed 2007-01-25).

[Riy07] Riya. Riya - Visual search. http://www.riya.com/, 2007. (accessed 2007-05-31).

[TN07] Tudorache and Noy. Collaborative Protégé. In *Workshop on Social and Collaborative Construction of Structured Knowledge, 16th International World Wide Web Conference (WWW 2007), Canada*, 2007.

[VKV+06] Völkel, Krötzsch, Vrandecic, Haller, and Studer. Semantic Wikipedia. In *15th international conference on World Wide Web. 2006, Edinburgh, Scotland, 2006*, 2006.

Semantic Integrator: Semi-Automatically Enhancing Social Semantic Web Environments

Steffen Lohmann, Philipp Heim, Jürgen Ziegler

University of Duisburg-Essen
Dep. of Informatics and Applied Cognitive Science
Lotharstrasse 65, 47057 Duisburg, Germany
{heim, lohmann, ziegler}@interactivesystems.info

Abstract: Large amounts of information from various sources have often to be considered when collaboratively developing semantic structures. Examining all relevant information can be very demanding and time consuming. Thus, methods and tools are needed that assist in the integration of this heterogeneous and distributed information. Based on an approach that uses Social Software and Semantic Web technology in requirements engineering, this paper describes the general concept and architecture of the Semantic Integrator, a tool that aims at visually support the integration of distributed information into semantic structures.

1 Introduction

The comprehensive collection of requirements is essential to successful software development. However, considering all sources of requirements and collect, analyze and merge the gathered information is challenging, particularly if the user groups are very large and spatially distributed. Semantic Web and Web 2.0 technologies open up new opportunities to better cope with these difficulties. Within the SoftWiki research project [Sof07], a web based collaborative environment is developed that fosters stakeholder participation in early requirements engineering. The SoftWiki philosophy follows the notion of the Social Semantic Web: Participation should be as easy as possible and semantically structured at the same time.

Though this "Wiki Way" [LC01] of requirements elicitation lowers the participation barrier and increases stakeholder involvement, large parts of stakeholders may still not have the skills, time, or motivation to actively use the collaborative environment. Furthermore, relevant information may already exist in some form or other but needs to be integrated. Examples are end user statements made in e-mails or webforms, on blogs or discussion boards, as well as existing documents and system descriptions. Thus, we search for ways to enhance Social Semantic Web Environments[1] by integrating these distributed information in an efficient way.

[1] By *Social Semantic Web Environments* we mean community platforms that combine Social Software and Semantic Web technologies (e.g. Semantic Wikis).

In the following we describe the general concept and architecture of the Semantic Integrator, a tool we are currently developing within the SoftWiki project. It aims at visually support the integration of information from diverse sources into an existing semantic structure (e.g. an ontology).

2 Semantic Integration

Three basic principles are at the heart of our approach: (1) The semantic integration should follow a semi-automatic process – manual and automatic activities shall complement each other. (2) The automatic integration should evolutionary improve by learning from the manual integration. (3) The integration process must always remain in the control of the user.

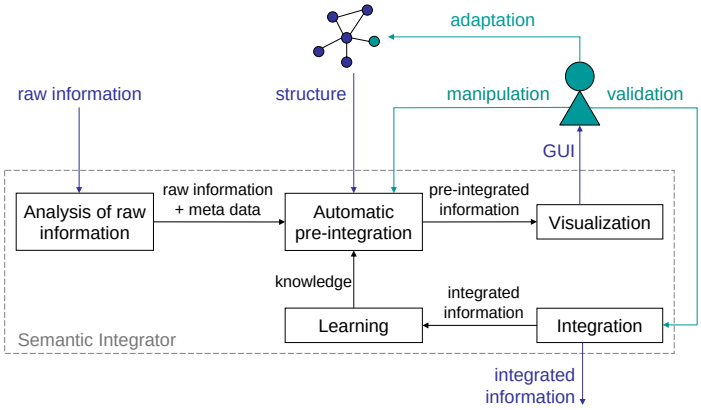


Figure 1: Raw information is analyzed and automatically pre-integrated in the existing structure. The result is then visualized to the user, who interactively manipulates and validates the pre-integration or adapts the underlying structure. Whenever information is manipulated or integrated this leads to a learning step for the next automatic pre-integration.

Usually, two kinds of input sources are provided to the Semantic Integrator (cp. Figure 1): an already existing semantic structure and one or more text documents containing the information that is intended to be integrated into the structure. We subsume the latter under the term *raw information* – though this information may be delivered in a structured way, it usually does not directly fit with the semantic scheme of the existing structure[2]. The Semantic Integrator aims to be able to process various XML-based input formats: The semantic structure may be provided in RDF or OWL, the raw information in XHTML or OpenDocument format. As output, RDF is generated that contains the adapted structure and the information integrated into it. With these standardized XML-based input and out-

[2]Furthermore, the Semantic Integrator may be used as a visual tool that assists in building an initial structure out of the raw information in cases where a semantic structure does not already preexist.

put interfaces, it will be possible to seamlessly plug the Semantic Integrator into Social Semantic Web Environments. The semantic integration process consists of the following components:

2.1 Analysis of Raw Information

Having selected the sources that should be considered for semantic integration, the included raw information is first analyzed. For this purpose, we use several text mining algorithms that work in conjunction with a large reference corpus [HQW02] and that have already been successfully applied in former research projects (see e.g. [ZJB05]).

First, the text is segmented into its single sentences and words, the stop words are eliminated and an index is generated. Typical word usage is derived by comparison with the reference corpus. Additionally, collocations are calculated and compared with the reference corpus. A collocation is the significant co-occurrence of two or more words within a well-defined unit of information (cp. [MS99]). The significant key words that are extracted out of the raw information in this process are then passed to the automatic pre-integration component.

2.2 Automatic Pre-Integration

To reduce the effort to integrate the raw information in the given structure as well as to extract or expand a structure out of the information, the system executes an automatic pre-integration step. In this step, the extracted key words are classified as far as possible according to the preexisting structure. For significant key words that cannot be assigned to any of the existing classes of the structure, suggestions for new classes are provided that might be valuable extensions to the structure.

To adequately integrate raw information in a certain structure we consider an automatic integration possible only to a certain extent. Hence, both the automatic classification of the key words as well as the extensions of the class structure are merely suggestions for the integration of raw information and need to get confirmed, manipulated or rejected by the user employing the Semantic Integrator GUI.

2.3 Visualization

The Semantic Integrator GUI is divided into three areas (cp. Figure 2): Firstly, a tree view, providing a hierarchical visual presentation of the preexisting structure plus the automatically derived suggestions for its extensions. Secondly, a similarity view, using a map-based visualization to show how the raw information is pre-integrated into this structure. And

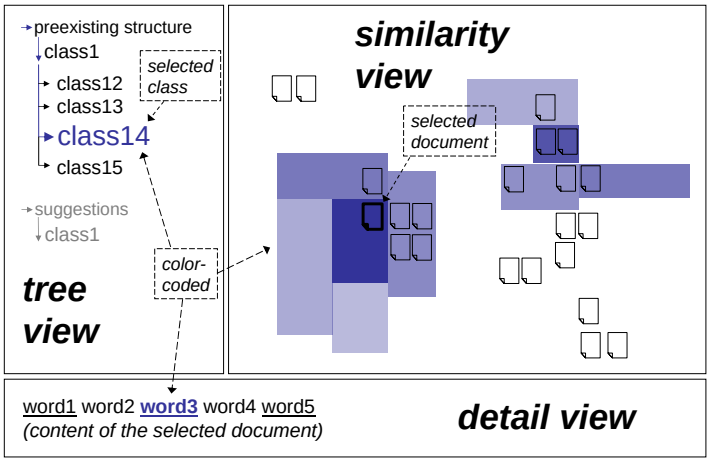


Figure 2: The Semantic Integrator GUI is divided into three areas: a tree view, a similarity view and a detail view.

thirdly, a detail view displaying details for a selected class from the tree view or a selected document from the similarity view.

In order to produce the similarity view we use the Vector Space Model [SAB94]. Hence, every distinct element of the raw information (i.e. every document), is represented by a vector of key words in a vector space spanned by the classes of the preexisting structure and the proposed extensions. The similarity of two documents can now be calculated by taking a similarity measure of the corresponding document vectors using either the scalar product, the cosine similarity measure or the Euclidean distance.

To provide a proper visualization of these similarities we use Self-Organizing Maps (SOMs) [Koh00] to scale our high-dimensional vector space onto a two-dimensional grid. This dimensionality reduction positions similar documents close to each other, which clusters the most related documents and thereby preserves the topology of the input vector space. Such a visualization of the pre-integrated raw information helps the user to identify similarities between documents and to get an overview of related topics.

In addition to the optimal organization of similar documents onto a two-dimensional grid using SOMs, the Semantic Integrator GUI provides color-coded information in accordance to the classification of the key words in the documents. If the user selects a class in the tree view, documents in the similarity view are color-coded depending on whether their key words are assigned to this class or not. If assigned key words are shown in the detail view, they get color-coded, too (cp. figure 2).

2.4 Integration

Equipped with the visualization of the clustered and pre-integrated raw information, the user can then either validate, reject or modify the given pre-integration. The same holds for the automatically generated extension of the class structure. The user can again validate, modify or reject the suggested new classes or build own extensions. We aim to provide intuitive interaction support that enables the user to integrate the raw information by selection, navigation, and drag&drop interaction.

2.5 Learning

The manual integration is then processed in a machine learning step to improve the pre-integration for the next cycle. The objective is to learn classifiers from integration patterns and enhance the quality of the pre-integration. This is implemented by Support Vector Machines (SVM) [Joa98], a supervised learning method that uses an efficient learning algorithm that can represent complex, nonlinear functions. The classification function for every class is learned by training data, in this case the manually integrated raw information. So every manual integration step, such as the handling of a pre-integration or an own classification, serves as training data for the SVM. Based on this data, the SVM calculates the optimal linear separator, a maximum-margin hyperplane, to classify unfamiliar information. Thus, every manual integration evolutionary improves the quality of the automatic pre-integration.

3 Conclusion and Future Work

The Semantic Integrator aims to serve as a semi-automatic tool for organizing, visualizing and integrating distributed information and adapting the underlying structures. Social semantic Web Environments benefit from this approach as information from diverse sources can be considered in the collaborative process in an efficient way. With respect to our requirements engineering approach, we will be able to consider information that is not directly expressed by stakeholders in the collaborative environment.

Future Work includes further development of the Semantic integrator and its incorporation in OntoWiki [ADR06], a tool for collaborative development of ontologies that is used in SoftWiki to gather requirements. Communication between the tools will be realized via REST and SPARQL. This incorporation will enable us to evaluate the achieved semantic integration within a use case in the context of the Social Semantic Web.

References

[ADR06] Sören Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In *Proceedings of the 5th International Semantic Web Conference*, pages 736–749, 2006.

[HQW02] Gerhard Heyer, Uwe Quasthoff, and Christian Wolff. Automatic Analysis of Large Text Corpora - A Contribution to Structuring WEB Communities. In *IICS '02: Proceedings of the Second International Workshop on Innovative Internet Computing Systems*, pages 15–26, London, UK, 2002. Springer-Verlag.

[Joa98] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with many Relevant Features. In *Proceedings of 10th European Conference on Machine Learning*, pages 137–142, 1998.

[Koh00] Teuvo Kohonen. *Self-Organizing Maps*. Springer, December 2000.

[LC01] Bo Leuf and Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley, 2001.

[MS99] Christopher D. Manning and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

[SAB94] Gerard Salton, James Allan, and Chris Buckley. Automatic structuring and retrieval of large text files. *Commun. ACM*, 37(2):97–108, February 1994.

[Sof07] SoftWiki. National research project funded by the German Federal Ministry of Education and Research (BMBF), 2007. http://softwiki.de.

[ZJB05] Jürgen Ziegler, Zoulfa El Jerroudi, and Karsten Böhm. Generating Semantic Contexts from Spoken Conversation in Meetings. In *IUI '05: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 290–292, 2005.

Semantic Wikipedia – Checking the Premises

Rainer Hammwöhner

Institut für Medien-, Informations- und Kulturwissenschaft
Universität Regensburg
Universitätsstraße
93040 Regensburg
rainer.hammwoehner@sprachlit.uni-regensburg.de

Abstract: Enhancing Wikipedia by means of semantic representations seems to be a promising issue. From a formal or technical point of view there are no major obstacles in the way. Nevertheless, a close look at Wikipedia, its structure and contents reveals that some questions have to be answered in advance. This paper will deal with these questions and present some first results based on empirical findings.

1 Introduction

Up to now Wikipedia has accumulated an enormous wealth of information by the effort of an open community of volunteers. This information however is semi-structured at best and therefore imposes restrictions on automatic processing. Automatic processing of Wikipedia contents is desirable for a couple of reasons. Enhanced information services can improve the utility of Wikipedia itself. Implicit knowledge scattered over separated parts of the corpus can be brought together and made explicit. Consistency of the corpus can be enforced by autonomous agents operating on semantic representations. Information extracted from Wikipedia can be used in other contexts.

There are several approaches to this task, but two very general types may be distinguished. The first approach employs information extraction from Wikipedia based on the interpretation of existing explicitly defined structures [AL07]. The main sources of information are templates embedded within Wikipedia's articles. The resulting knowledge is represented in terms of a formal language and may be subject to viewing and querying via the OntoWiki software [ADR06]. The second approach requires a modification of the markup language in order to allow for link typing and attribute assignment [Vö06]. A process of information extraction and representation will again lead to formal representations that may be employed by inference processes. A necessary prerequisite of this approach is an extension to the MediaWiki software that is the technical core of Wikipedia [KVV06].

According to [Vö06] the following key elements are necessary to achieve the intended semantic annotation of Wikipedia's articles: *categories* classify articles according to their content, *types* express the meaning of links connecting Wikipedia's articles and *attributes* capture atomic properties related to the contents of an article. Categories are the only of these devices being already in use and ready for evaluation. Thus the notion of categorizing Wikipedia's articles will play a crucial role within the theoretical and practical considerations of this paper.

2 The Premises

Introducing at least one of the approaches mentioned above will be of major consequence to the users of Wikipedia. New information services will be available on the one hand and the authoring process will be more demanding on the other hand. The success of this project is bound to some central premises that should be made explicit and checked before the effort of large scale implementation is to be taken.

P1 Technical feasibility: Prototypes for both of the approaches have been implemented.

P2 Formal soundness: The proposed semantic representations are based on rigidly defined structures. However, there is some lack of clarity about the further use of typed links. As far as no terminological reasoning is intended, no problems should arise.

P3 Reliability of results: Recent studies have attested Wikipedia's convenient average quality [Ha07a, Ha07b, Wi07]. However, Wikipedia articles of abysmal quality can be found easily. The user of Wikipedia needs the competence to distinguish reliable from erroneous information. Semantic operations on Wikipedia should not accumulate errors and must not blur the user's view by hiding the sources of errors. It is not quite clear, whether this criterion is met by the proposed approaches.

P4 Reliability of the authoring process: The first approach does not impose additional tasks on the author. No new problems should arise here. The second approach relies heavily on the proper assignment of link types and categories by the user. The author can decide which and how many link types or categories to use. He can select from predefined denominators or enter new link types and categories at his will. Obviously, problems can arise out of the inconsistent and ambiguous use of type and category identifiers. [Vö06, section 4.1] infer from the seemingly unproblematic use of the category system that a consistent use of a link type system is to be expected too. This conclusion is problematic simply, because there is no empirical evidence of a proper use of the category system at all. It is the major objective of this paper to present some observations which are relevant to this issue.

P5 Multi-lingual system: Approaches to realizing a Semantic Wikipedia should consider that Wikipedia is a multi-lingual information base. At least an interlingual mapping mechanism for link types and attributes corresponding to interlingual category mapping should be developed.

P7 Usability: All efforts in enhancing Wikipedia by innovative information services will be futile unless they are integrated within an environment devoted to strict usability criteria. This applies for the authors and information seekers as well.

The list introduced above may not be complete. But the relevance of the mentioned premises does not seem to be questionable. ***P4*** occupies a key position since a fundamental question is involved here. Usable interfaces may be revamped, formal systems can be redesigned, but the competence of a large user community can be adjusted only in the long run. Thus ***P4*** may be the decisive criterion in the choice between more or less demanding approaches to a Semantic Wikipedia.

3 Is Wikipedia's category system a sound thesaurus?

The category system of Wikipedia is intended to provide an additional navigation structure on the set of articles [Wi07a]. It is not used as a device of query support primarily. The proper assignment of categories is defined by a set of rules of thumb [Wi07a]. The question, whether this category system is a thesaurus, was firstly brought up by [Vo06]. In his comprehensive overview on the category system of Wikipedia Voss examines the statistical distribution of category features and compares this category system to other means of knowledge organization - thesauri (MeSH: Medical Subject Headings), hierarchical classifications (Dewey Decimal Classification) and folksonomies (del.icio.us). This comparison is of major importance to Semantic Wikipedia, because formal properties of the category system may be inferred from the result. Voss arrives at the conclusion, that Wikipedia's category system is a thesaurus, since the requirements of ISO 2788 [ISO86] are met. The *equivalence relation* connecting synonymous terms may be represented using redirects. The *hierarchical relation* between broader and narrower terms is expressed by the category ⇒ subcategory relation. *Associations* between related terms are represented by hyperlinks. Obviously the mark-up language of Wikipedia is capable of expressing thesaurus structures. The question, however, is, whether the existing category systems *are* thesauri. [Vo06] further elaborates his conclusions by comparing excerpts from the MeSH thesaurus and from the English Wikipedia. The presented structures are reasonably similar. But counter examples may be found easily at least within the English Wikipedia (as observed at. 0.6.07.2007):

> *categories* ⇒ *fundamental* ⇒ *thought* ⇒ ***knowledge*** ⇒ *academia* ⇒ *academic institutions* ⇒ *school counseling* ⇒ *personal development* ⇒ *personal finance* ⇒ *microeconomics* ⇒ *information, knowledge and uncertainty* ⇒ *information* ⇒ ***knowledge*** ⇒ *nature* ⇒ *life* ⇒ *death* ⇒ *extinction* ⇒ *fossils* ⇒ *dinosaurs*

This illustrative example demonstrates the existence of cycles (*knowledge*) within the category ⇒ subcategory relation. Cyclic structures conform to Wikipedia's rule set [Wi07a], but not to ISO 2788 since the resulting structure is no hierarchy. The category ⇒ subcategory relation does not lead generally from broader to narrower terms, but in many cases to related terms. Thus, the category ⇒ subcategory relation may not be considered as a transitive relation representing terminological subordination.

As a consequence there is no support of terminological reasoning by the English category system. Even retrieval support, e.g. by spreading activation, may lead to unwanted results, if the terminology is as weakly structured as the example suggests. The same criticism is valid for the French Wikipedia as well. The category systems of the Italian and German Wikipedia are quite different in structure. They contain a few cycles only, their hierarchy has a considerably lower depth (s. table 1). This applies to the maximal descriptor level (first value) and the longest observed path within the hierarchy (value in brackets) as well. A substantial difference between both of the depth values indicates a lack of balance within the category system.

	articles	basic categories	all categories	max depth	superord. per cat. (median)	cycles
de (en)	152	366	1740	10 (15)	2	4
de (fr,it)	169	394	1816	10 (15)	2	4
en	152	581	6274	14 (156)	2	493
it	167	321	1091	12 (15)	2	7
fr	134	360	3116	14 (83)	2	424

Table 1: Basic features of category systems

The data presented above are derived from the following samples: two bilingual samples of de-en (size 152) and de-it (size 169) were chosen at random using interlingua links. A sample of 134 French articles was added to the latter one, once more using interlingua links. The basic categories describing these articles were sampled as well as all of their superordinate categories. The example suggests, that sample size has some influence on the number of basic categories, less influence on the total number of categories and no impact on the depth of hierarchy and number of cycles. It can be assumed, that deep category systems are error prone. Authors will have difficulties to get an overview on the overall structure since the number of paths to the top category shows exponential growths behaviour. An additional example will illustrate the pitfalls of big category hierarchies in Wikipedia. It shows the first 99 categories of the longest path within the category ⇒ subcategory multi-hierarchy as found in the sample of the English Wikipedia:

digital revolution ⇒ cryptography ⇒ application of cryptography ⇒ authentication methods ⇒ personal identification ⇒ biometrics ⇒ physical anthropology ⇒ human evolution ⇒ evolutionary psychology ⇒ memetics → anticipatory thinking ⇒ strategic management ⇒ product management ⇒ product development ⇒ design ⇒ built environment ⇒ architecture ⇒ architecture and engineering occupations ⇒ building engineering ⇒ building materials ⇒ metals ⇒ alloys ⇒ copper alloys ⇒ bronze ⇒ bronze age ⇒ ancient near east → ancient near eastern religions → ancient semitic religions ⇒ Abrahamic religions ⇒ Judaism → messianism ⇒ Jesus ⇒ doctrines and teachings of Jesus → nonviolence → peace ⇒ peace churches ⇒ anabaptism ⇒ amish ⇒ simple living ⇒ environmentalism ⇒ environmental ethics ⇒ extinction ⇒ extinct species

⇒ extinct animals ⇒ prehistoric animals ⇒ mesozoic animals ⇒ cynodonts ⇒ mammals⇒ primates ⇒ apes ⇒ humans ⇒ anthropology ⇒ prehistory ⇒ archaeology ⇒ periods and stages in archaeology ⇒ ancient history ⇒ ancient mysteries ⇒ astrology → ~~astrological factors~~ → classical elements → earth⇒ earth sciences ⇒ environmental science ⇒ environment ⇒ urban studies and planning ⇒ transportation ⇒ travel ⇒ tourism ⇒ cultural heritage ⇒ cultural history ⇒ cultural movements ⇒ art genres ⇒ graphic design ⇒ printing ⇒ books → fiction ⇒ fictional ⇒ fictional abilities ⇒ superhuman powers ⇒ psychic powers → prediction ⇒ futurology ⇒ population ⇒ demography ⇒ ethnicity ⇒ ethnicity in politics ⇒ anti-national sentiment ⇒ prejudices ⇒ bias ⇒ appearance ⇒ aesthetics ⇒ arts ⇒ visual arts → communication design ⇒ mass media ⇒ media by format ⇒ digital media ⇒ software ⇒ software engineering ⇒ ...

This example was extracted from the English Wikipedia at 15th of June and verified at the 7th of August 2007. In the meantime one category and 10 category ⇒ subcategory links have been deleted (→), a super-category has been added to *digital revolution* again. Some of these deletions lead to a simplification of the overall structure; some others were caused by the insertion of additional hierarchy levels. It is an open question, which effects will result from the volatility of the category system as observed in this example. These findings, however, have to be confirmed using bigger samples or the complete data set. It would be desirable to develop diagnostic tools which could identify problematic category inclusions. One promising approach is the comparison of category systems from various Wikipedias. If a category ⇒ subcategory inclusion is present in more than one Wikipedia, it is likely to be valid. If it occurs in one Wikipedia only, it can be invalid or culture specific as well.

4 What does this mean to Semantic Wikipedia

This small study, based more on illustrative examples than on statistical evidence, suggests that Wikipedia's category system is not obviously a sound base for the development of a more demanding semantic system. The proliferation of the category system indicates what may happen to a link type system that may freely be extended by the user. This aspect is of crucial importance since evaluation of link typing had controversial results even in more controlled settings [Ma91]. As a consequence more empirical studies on category assignment are needed in order to understand the unfolding of the rather different category systems within the German and Italian Wikipedia on one side and the French and English Wikipedia on the other. Various settings – for instance with open and closed link type systems – should be considered before modifications at the existing encyclopaedia are brought into effect. Nevertheless, the introduction of more semantic features into Wikipedia has lots of promising aspects, too. The category system can be relieved from alien tasks like fact representation. The problem of redundant assignment of categories and subcategories [Wi07b] to Wikipedia articles can be solved by simple inference processes in combination with appropriate presentation tools. These are just examples of the positive effects that can be achieved by Web 2.0 techniques. Furthermore, the technical soundness and good performance of the existing prototypes promises that experiments may be carried out with reasonable effort.

References

[AL07] Auer, S.; Lehmann, J.: What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. accepted at ESWC 2007. http://www.informatik.uni-leipzig.de/~auer/publication/ExtractingSemantics.pdf, cited 20.05.2007

[ADR06] Auer, S.; Dietzold, S.; Riechert, T.: OntoWiki - A Tool for Social, Semantic Collaboration. In I. Cruz et al. (Eds.): Proceedings of 5th International Semantic Web Conference, Nov 5th-9th, Athens, GA, USA, LNCS 4273, pp. 736-749, 2006. Springer-Verlag Berlin Heidelberg 2006. http://www.informatik.uni-leipzig.de/~auer/publication/ontowiki.pdf, cited 20.05.2007

[Ha07a] Hammwöhner, R. et.al.: Qualität der Wikipedia. Eine vergleichende Studie. In Oßwald, A.; Stempfhuber, M; Wolff, C. (eds.) Open Innovation. Proc. 10th Int. Symposium on Information Science in Cologne. UVK, 2007, pp. 77-90.

[Ha07b] Hammwöhner, R.: Qualitätsaspekte der Wikipedia. In: Stegbauer, C.; Schmidt, J.; Schönberger, K. (eds): Wikis: Diskurse, Theorien und Anwendungen, Sonderausgabe von kommunikation @ gesellschaft, Jg. 8, 2007, Online-Publication: http://www.soz.uni-frankfurt.de/K.G/B3_2007_Hammwoehner.pdf

[ISO86] ISO 2788: 1986: Guidelines for the establishment and development of monolingual thesauri.

[KVV06] Krötzsch, M.; Vrandecic, D.; Völkel, M.: Semantic Mediawiki. In Proc. 5th Int. Semantic Web Conf. (ISWC06). http://korrekt.org/papers/KroetzschVrandecicVoelkel_ISWC2006.pdf, cited 20.05.2007

[Ma91] Marshall, C.C. et.al.: Aquanet: a hypertext tool to hold your knowledge in place. In *Proc. Hypertext' 91, San Antonio,* S. 261-275, New York, 1991. ACM.

[Vö06] Völkel, M. et.al.: Semantic Wikipedia. In Proc. 15th Int. Conf. on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006. http://www.aifb.uni-karlsruhe.de/WBS/hha/papers/SemanticWikipedia.pdf, cited 20.05.2007

[Vo06] Voss, J.: Collaborative thesaurus tagging the Wikipedia way. (v2; 2006-04-27; http://arxiv.org/abs/cs.IR/0604036) – Wikimetrics research papers, volume 1, issue 1 (cited 0.6.07.2007).

[Wi07] Wiegand, D.: Entdeckungsreise, Digitale Enzyklopädien erklären die Welt, c't, Magazin für Computer und Technik, Nr. 6, 2007, S. 136-145.

[Wi07a] Wikipedia:Categorization. In Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Wikipedia:Category, cited 27.05.2007.

[Wi07b] Wikipedia: Categorization and subcategories. In Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/Wikipedia:Categorization_and_subcategories, cited 27.05.2007.

[Wi07c] Wikipedia Diskussion: Kategorien. In Wikipedia, The Free Encyclopedia. http://de.wikipedia.org/w/index.php?title=Wikipedia_Diskussion:Kategorien/Archiv1, …Archiv2, …Archiv3, cited 27.05.2007

Exploring the Netherlands on a "Semantic Path"

Michael Martin

Business Information Systems, University of Leipzig
o-mega@gmx.de
Postfach 100920
04009 Leipzig

Abstract: This poster gives an overview about the web application "vakantieland.nl", a Dutch Internet portal in the tourism domain. The core functionality is to provide information about holiday destinations, accommodation and other tourism related points-of-interest as well as a corresponding visualization with a mapping service. Because of the underlying semantic data structures and alternatively generated RFD output, vakantieland.nl can be considered to be a Semantic Web application. Its realization and functionalities, social aspects and furthermore an outlook about future development work constitute the main part of the poster.

1 Introduction

Semantic Web [BHL01] and Web 2.0 [Or05] are two concepts that expanded and redefined the possibilities of the Internet. The Semantic Web expands the World Wide Web with ability to represent information in a machine readable way by formally defining the Semantics of the content. Web applications that use such technologies are called Semantic Web applications. To further integrate the user into the web application's processes, an extension of functionalities, which come from the ideas of Web 2.0, is necessary. These Web 2.0 functionalities serve to expand and improve the existing knowledge base and allows many different views on a resource.

Many web applications that were created in an evolutionary process can not be adapted with little effort to new techniques, technologies and use cases. The only way in most cases to guarantee an up-to-date appearance is to completely redesign the application. Vakantieland.nl was one of those cases. The amount of added data that needed to be displayed and the huge collection of functionalities that had to be developed, formed the decision to completely redesign the application. While in the second section the original design of the application is presented, section three then describes the new version and the changes made. In section four we describe the included social aspects based on Web 2.0 and finally the fifth section gives an outlook about functionalities that still need to be developed.

2. The original application

The previous version of vakantieland.nl displayed tourism destinations (e.g. destinations and information points in cities), which are called “Point of Interest” (“POI”). Every POI was associated with at least one category and could be viewed either by selection of the category or through other relations, it had, to other POIs. The output of POIs were in the form of lists and points on maps. The maps, where those points were displayed upon, were saved as images in different zoom steps. Then the tourism destinations where positioned on the maps with the help of their coordinates. For every POI information like name, contact addresses, opening hours and pictures was provided and could be viewed in detail next to the map as can be seen in Figure 1. Additionally a map of the area showed a list of close-by POIs. These associations served as a further mean of navigation between POIs.

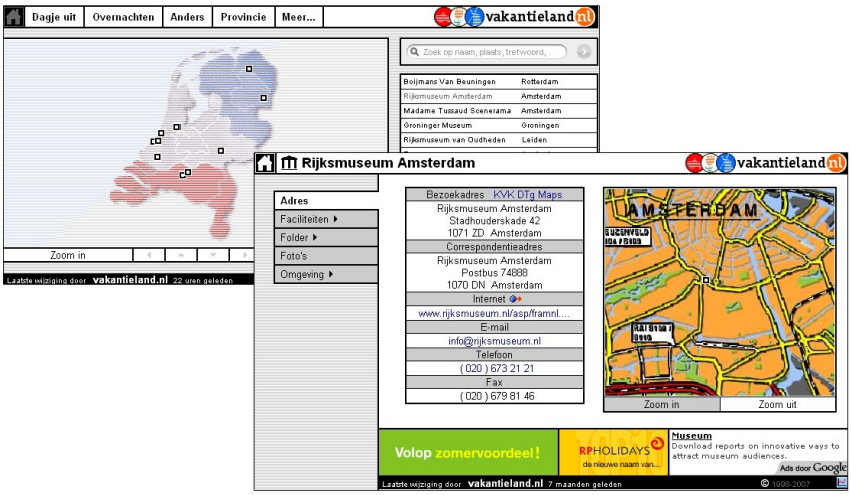


Figure 1: GUI of the previous version of the vakantiland.nl portal

The data of this application was kept in a relational data base and were retrieved with ASP[1] to display them. Furthermore filtered information was extracted with VisualBasic and saved in XML[2]. This extracted data was used in external pages.

3. The new version of the vakantieland.nl portal

The company of the portal required the usage of forward-looking technologies to provide a better overview, improved usability and more diverse information on the one hand and to allow the employees an easier handling and maintenance of the site on the other hand. The user also was supposed to be integrated and should have the ability to interactively

[1]ASP : **A**ctive **S**erver **P**ages, http://asp.net/
[2]XML: e**X**tensible **M**arkup **L**anguage

create and expand the knowledge base. All data should be made shareable in a machine-readable way, so it could be processed by other applications and sites. To fullfill all the above mentioned requirements the use of Semantic Web technologies with exchangeable data formats and standardized namespaces and APIs, which allow the processing thereof, was necessary.

The scheme and the data of the model of vakantieland.nl was created and is kept in RDF [Be04]. RDFS[3] or OWL[4] enables us to define every conceptual part of the data model (like classes and properties) and the related instances as resources with the corresponding Resource Identifier (URI). Displayed categories are modelled in a class tree. This serves to allow sub classification of categories. For example the category *Hotel* contains the subcategories *ApartmentHotel, EconomyHotel, CastleHotel*, etc. In the model, these subcategories are subclasses of the class Hotel, as can be seen in Figure 2.

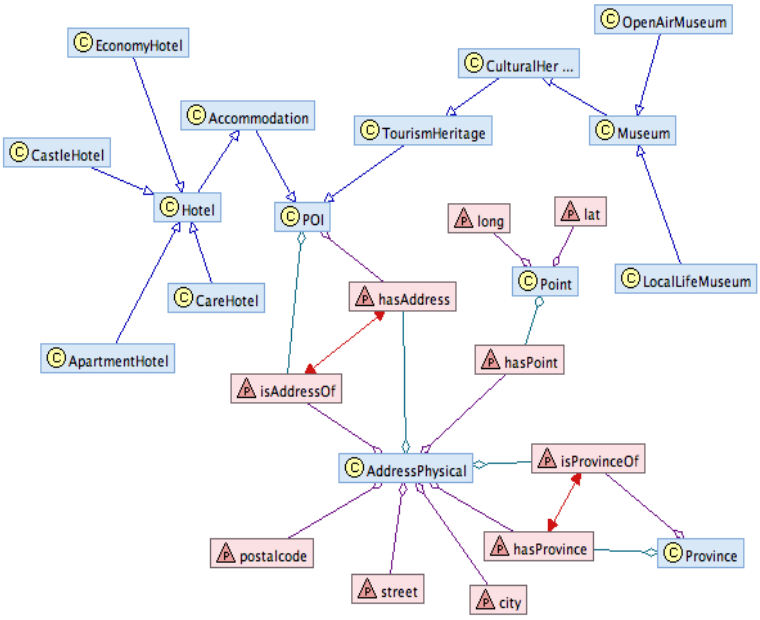


Figure 2: Excerpt of the model

To assign POIs to different categories polymorphism is necessary using this way of modelling. To give an example, there could be a special hotel which also has an public restaurant, which belongs to another category. To make the model accessible by the the application, it was saved in a database-backed TripletStore [MD00]. For this the Powl-

[3]RDFS : http://www.w3.org/RDF/
[4]OWL : http://www.w3.org/2004/OWL/

API[5] [Au05] was used, which also handled the necessary transformation. To verify if the resulting model and the used SPARQL[6]-queries [Be06] were reusable, we tested it with Ontowiki [ADR06].

The structure of the application, which is implemented in the script language PHP[7] follows a well known architectural pattern called MVC[8] [Bu98]. The model component abstracts the accesses by the controller on the Powl-API, which in this case also manages the extraction of data from the TripletStore to make the data suitable for the requirements. The RAP[9]-API [OB04] which is used by the Powl-Api allows to query the model with SPARQL [Be06]. SPARQL was used to identify and find resources with specified characteristics in an efficient manner. After the identification of the resource URI, an object with all the existing information is created with the Powl-API. Such generated objects will be requested by the appropriate controller and assigned to the view component. The view component produce the output with the help of templates.

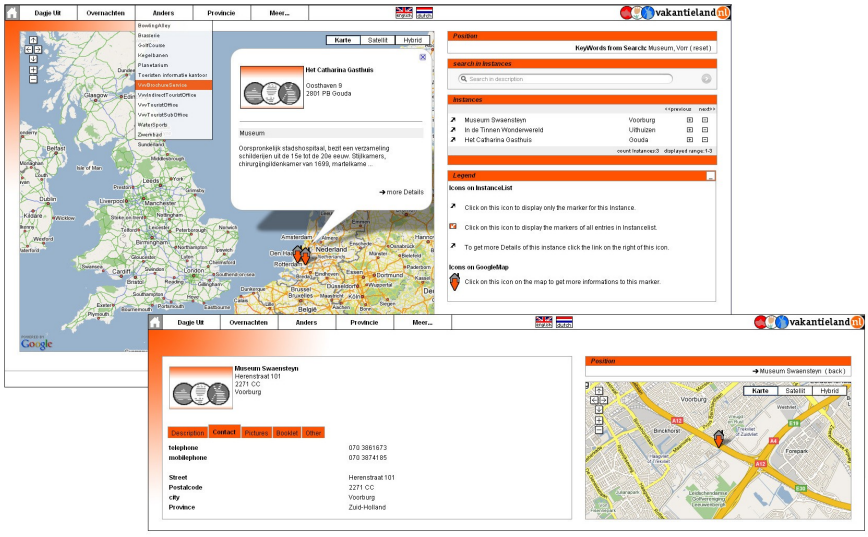


Figure 3: GUI of the new vakantieland.nl portal

The design of the templates was chosen to appear similar to the previous version, but the information and the namespaces are anchored in the HTML[10]-code to make it machine-readable and allow processing by other applications. Thus, the title and the description of a POI for example are marked Dublin Core[11] properties. The POIs are positioned and

[5]Powl-API: http://powl.sourceforge.net/
[6]SPARQL: recursive Acronym for „**S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage“
[7]PHP : recursive backronym for „PHP: **H**ypertext **P**reprocessor“ http://www.php.net
[8]MVC: **M**odel **V**iew **C**ontrol
[9]RAP: **R**DF **A**PI for **P**HP
[10]HTML: **H**yper **T**ext **M**arkup **L**anguage
[11]Dublin Core : http://dublincore.org/

displayed on the map by using the GoogleMaps-API[12] as you can see in Figure 3. The API is used to retrieve the coordinates of the POIs and also to display the map material in conjunction with the related map-markers.

4. Social Semantic Web in vakantieland.nl

Another requirement for the new version vakantieland.nl is to achieve a higher degree of user interactivity. In the previous version of vakantieland.nl users only had the possibility to get informations about POIs. One way to improve the user interactivity is the addition of more functionalities to collect user generated contents. Thus users get the possibilities to share there experiences now.

Therefore new functions were added like the evaluation of POIs and feedback abilities to grade the up-to-dateness of the data and the user now has the possibility to write textual comments like in a guest book. Through this, other interested users now have the advantage to profit from those shared experiences and personal opinions. Furthermore users can now make statements about the actuality and validity of data, which supports the administrators of the application and minimizes the effort to keep the data up-to-date. A well defined, closed user group (registered users with the role 'author') now has the ability to change data like the description of a POI or the opening hours directly.

In this manner interested people that want to explore the netherlands can do it now on an easy semantic path.

5. Outlook

Finally, we give a short description about planned, but not yet implemented functionalities to further improve the diversity of available information and the ability to interact with the application. Users might wish more information than is provided by the knowledge base of the vakantieland.nl project. It is possible, that they, e.g. in search of a museum in Amsterdam, might not be only interested in opening hours, but also want to know more about the history of the museum or even the history of the city, the museum is situated in. To provide for this demand queries are send to a SPARQL-Endpoint of the DBpedia[13] portal to get more background information about the city, the POI itself or other relevant issues. The results can be integrated in already displayed information. Finally, web services are planned, which supply data in different formats like JSON[14] and RDF to requesting clients.

[12]GoogleMaps-API: http://www.google.com/apis/maps/
[13]Dbpedia : http://dbpedia.org
[14]JSON: **J**ava**S**cript **O**bject **N**otation

References

[Au05] Sören Auer : Powl – A Web Based Platform for Collaborative Semantic Web Development, Scripting for Semantic Web, Vol-135, 2005

[ADR06] Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki - A tool for social, semantic collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, volume 4273, pages 736-749. Springer, 2006.

[Be04] Dave Beckett : RDF/XML Syntax Specification (Revised). W3C. 10February 2004.

[Be06] Dave Beckett : SPARQL RDF Query Language Reference V1.8 2006 http://www.dajobe.org/2005/04-sparql/SPARQLreference-1.8.pdf

[BHL01] Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web. Scientific American 284 (2001) 35–43

[Bu98] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal. Pattern orientierte Softwarearchitektur, Addison Wesley 1998

[MD00] Sergey Melnik, Stefan Decker: A Layered Approach to Information Modeling and Interoperability on the Web 2000 http://infolab.stanford.edu/~melnik/pub/sw00/

[OB04] Radoslaw Oldakowski, Christian Bizer - RAP: RDF API for PHP, 1st International Workshop on Interpreted Languages (colocated with the NetObjectDays 2004 Conference), Erfurt, September 2004

[Or05] Tim O'Reilly – What is Web 2.0, 2005, http://www.oreilly.com/