SCALABLE COMPUTING
Practice and Experience

Special Issue: The Web on the Move

Editors: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz

Volume 9, Number 4, December 2008

ISSN 1895-1767

UNIVERSITÄT SALZBURG

UNIVERSITATEA DE VEST DIN TIMISOARA

SUBSCRIPTION INFORMATION: please visit http://www.scpe.org

Scalable Computing: Practice and Experience

Volume 9, Number 4, December 2008

TABLE OF CONTENTS

Scalable Computing: Practice and Experience
Volume 9, Number 4, p. i. http://www.scpe.org
ISSN 1895-1767

EDITORIAL

Dear Readers!

This is the last time I am communicating with you as an Editor-in-Chief of SCPE. A number of changes concerning our journal took place in 2008. First, already in late 2007 we stopped being published (supported) by the SWPS. Thus we continued working as a volunteer-based effort, while the SWPS graciously allowed us to use their server to host the journal. In the meantime we have worked on regaining control over the SCPE domain. It turned out that the ISP that we were using is extremely bureaucratic and this, combined with my usual overloaded schedule, meant that it took almost a year to be able to get the right documents, with the right signatures, within a specific time interval, delivered to the ISP office, so that they can finally process the change of domain ownership. I am glad to announce that we are now in full control of the scpe.org domain. In the meantime we have found a new home for the journal and a new sponsor. The SCPE is now housed at the University of Western Timisoara (in Timisoara, Romania) and is co-sponsored by the University of Salzburg (Salzburg, Austria). I would like to say that I am extremely grateful to Professor Dana Petcu of the University of Western Timisoara and to Prof. Marian Vajtersic, of the University of Salzburg for their support for our journal!

With the journal out of Poland, I have looked back and reflected on the past, present and future perspective of the SCPE. After some deliberation I came to the conclusion time has come for a change in the journal leadership. After a few long conversations, Professor Dana Petcu of the University of Western Timisoara has agreed to take over the post of SCPE Editor-in-Chief. At the same time we have came to the conclusion that it would be a good idea to create a Steering Committee of the journal and this new body will consist of: Dana Petcu, Marian Vajtersic, and myself.

I would like to use this occasion to express my deepest gratitude to all past and present members of the Editorial Board of the SCPE (and the PDCP, its predecessor). Without your hard work and support, the idea of the journal would have never materialize and came to the fruition. Thanks to you we were able to create a very good quality scientific journal that is now entering its 10th year of publication.

I would also like to thank dr. Alexander Denisjuk of the Elbląg University of Humanities and Economy, for the absolutely fantastic job that he is doing as the Managing and Technical Editor of the SCPE. He is the one who (re)designed the SCPE WWW site, fought with authors to get material prepared in the right way to produce professional quality issues and who fought with me to make sure that I do things on time and that we do not fall behind. Thank you, Alexander!

Sincerely Yours,
Marcin Paprzycki

Scalable Computing: Practice and Experience
Volume 9, Number 4, p. iii. http://www.scpe.org
ISSN 1895-1767
© 2008 SCPE

INTRODUCTION TO THE SPECIAL ISSUE: WORLD WIDE WEB ON THE MOVE

The papers collected in this issue present a wide range of research that can be connected to the three main topics: the Web, Knowledge and Social Networks. Each and every of the papers touches at least two of these areas to a certain extent.

The evolution of Web infrastructure, methods of representing information published on, and functionalities available is discussed in "World Wide Web on the Move" and gives a good overview of different aspects of today's Web on the move. A number of these aspects are further investigated.

In particular weblogs, Wikipedia, Deep Web and Rich Internet Applications are discussed in subsequent articles. They are the phenomena that attract a lot (if not most of) interest of contemporary research in the area that was recently named by the Web originator – Tim Berners-Lee – the Web Science[1]. Our Authors follow the trend by investigating: weblogs influence on Small and Medium Enterprises as means of conveying knowledge about management and the enterprise (paper by Alexander Stocker, Markus Strohmaier, and Klaus Tochtermann), Wikipedia as a knowledge-rich resource which might be used to annotate multimedia content (article by Angela Fogarolli and Marco Ronchetti), Deep Web as potentially ground-breaking information source, yet largely unexplored due to theoretical and practical limitations (contribution of Yang Wang and Thomas Hornung) and RIA as a new paradigm of exposing functionality on the Web combining state-of-the-art engineering paradigms and knowledge about an enterprise (for Kay-Uwe Schmidt, Roland Stuehmer and Ljiljana Stojanovic this knowledge comes in a form of business rules).

Social aspect of the Web Science was also studied in this issue. The work of Celine Van Damme, Tanguy Coenen and Eddy Vandijck describes the method of using socially created folksonomies to formalize enterprise knowledge into an ontology. Raf Guns exercises well known (although not entirely standard) social network analysis tools on social Semantic Web. Paolo Massa, Kasper Souren, Martino Salvetti and Danilo Tomasoni propose an open platform to exchange trust metrics and algorithms to compute them, which are crucial for social network adoption and prosperity.

Finally, the works that focus on knowledge subject include the paper by Dimitris Bibikas and his colleagues, that builds on a case study to propose novel approach to knowledge management within enterprise using Web paradigms, as well as the article by Alsayed Algergawy, Eike Schallehn and Gunter Saake describing fuzzy approach to solve a well known (and hard) knowledge representation matching problem in the context of Deep Web sources.

Following the traditional division into theoretical and applied approaches, in this issue there are papers that present theoretical developments and new methods (as in "Fuzzy Constraint-Based Schema Matching Formulation," "Discovering Semantics In Multimedia Content Using Wikipedia," "Deep Web Navigation By Example") or applied research in the form of blueprints for systems design and architectures (like "From Business Rules To Application Rules In Rich Internet Applications"), as well as empirical evidences ("Unevenness In Network Properties On The Social Semantic Web," "Studying Knowledge Transfer With Weblogs in Small and Medium Enterprises: An Exploratory Case Study" and "A Sociotechnical Approach To Knowledge Management In The Era Of Enterprise 2.0: The Case Of Organik"). As usual there are border cases, like "Deriving A Lightweight Corporate Ontology Form A Folksonomy: A Methodology And Its Possible Applications" and "Trustlet, Open Research On Trust Metrics," where Authors discuss both the method and its possible application scenarios.

Overall, the wide coverage of paper topics shows clearly, that the contemporary Web is both a field to apply old and well tested techniques to new problems, and a fertile environment for innovation. We sincerely hope that our Readers find among the work gathered in this issue an inspiration for new inventions and theories.

Dominik Flejter, Tomasz Kaczmarek,
Poznań University of Economics

Marek Kowalkiewicz,
SAP Research Brisbane

[1] http://webscience.org/

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 219–241. http://www.scpe.org
ISSN 1895-1767
© 2008 SCPE

WORLD WIDE WEB ON THE MOVE

DOMINIK FLEJTER[*], TOMASZ KACZMAREK[*], AND MAREK KOWALKIEWICZ[†]

Abstract. In this paper we provide an overview of key changes that happened on the Web in a few recent years. We start by analyzing changes occurring at the level of widely understood Web infrastructure (standards, computing, storage). Then, we focus on machine-oriented and user-centric trends in representation of information (both structured and unstructured). Next, we briefly discuss evolution of types of on-line functionalities and their access modes. Fourth component of the Web that we analyze is related to a few directions in actual usage of Web and its impact on social life. Final part of this paper is devoted to topics that span previous components such as driving forces, business models and privacy.

Key words: World Wide Web, evolution, Web infrastructure, Web data, Web resources, social Web, Web usage, business models

1. Introduction. World Wide Web is not only the biggest information repository in the history of humanity; it is also a dynamic and very quickly evolving universe consisting of people, businesses, applications, infrastructures and resources dynamically interacting with each other. This evolution results in an increasing complexity both of its individual components, and of the ecosystem formed by interplay of these elements. This paper provides an overview of evolving components of modern Web, mostly focusing on changes that happened in few last years.

1.1. Components of Evolving Web. In this paper we analyze evolution of the Web along four major areas (see Fig. 1.1). The first area (infrastructure), concerned with basic components and services that enable functioning of World Wide Web, is discussed in Section 2. It includes both hardware and software that enable different models of Web-based storage and computing, as well as basic standards (including file formats and communication protocols) that make Web-based communication possible. The next component, discussed in Section 3, concerns resources on the Web. It is mostly concerned with actual presence and representation methods of different kinds of Web content and data. The next area, covered in Section 4, focuses on functionalities available on the Web. It is concerned with the operations that users can perform on Web data and content, how they are accessible and how they can be combined. Finally, the forth component of proposed schematic view, described in Section 5, is related to usage scenarios of on-line systems. It focuses on which available functionalities people and businesses really use, and how important is their role in todays economical and social life.

These four components concern four distinct areas of contemporary World Wide Web. However, important interactions between them can be observed, as described in Section 6. They concern both driving forces of Web development, and issues that span multiple components, such as business models and privacy.

2. Infrastructure. At the very dawn of Internet its infrastructural level consisted mostly of wires and basic communication protocols and standards (such as DNS). Application protocols and data transfer formats were at their infancy, rather foreseen than fully developed. Over time it covered more complex components of Internet communication. Firstly, a number of standards of growing complexity such as HTML, JavaScript, CSS, XML, RDF and RSS appeared and became popular. Secondly, on-line documents storage became easier with no need to possess own servers: FTP and HTTP servers (including free options) became available to all Internet users and a number of alternative storage platforms (including blogs, on-line file sharing and social networking sites) became part of infrastructure. Thirdly, some basic Web computing platforms (e.g. Apache/MySQL/PHP, Python, RubyOnRails, ASP.NET) became omnipresent making deployment of Web applications easier.

Infrastructure is important as its availability at affordable rates or at no (direct) cost at all is one of the building blocks of all Internet activities. At some level of abstraction, we can perceive infrastructure as a large scale mechanism of demand accumulation to obtain economies of scale. As infrastructural components are required by everyone on the Web, keeping them shared by all makes technological and economical optimization possible. As the result of infrastructure availability, the entry barriers for new innovative business and social solutions acting on the top of them are lowered.

[*] Poznan University of Economics, Department of Information Systems, al. Niepodleglosci 10, 60-967 Poznan, Poland, {D.Flejter, T.Kaczmarek}@kie.ae.poznan.pl

[†] SAP Research CEC Brisbane, Level 12, 133 Mary Street, Brisbane QLD 4000, Australia, marek.kowalkiewicz@sap.com

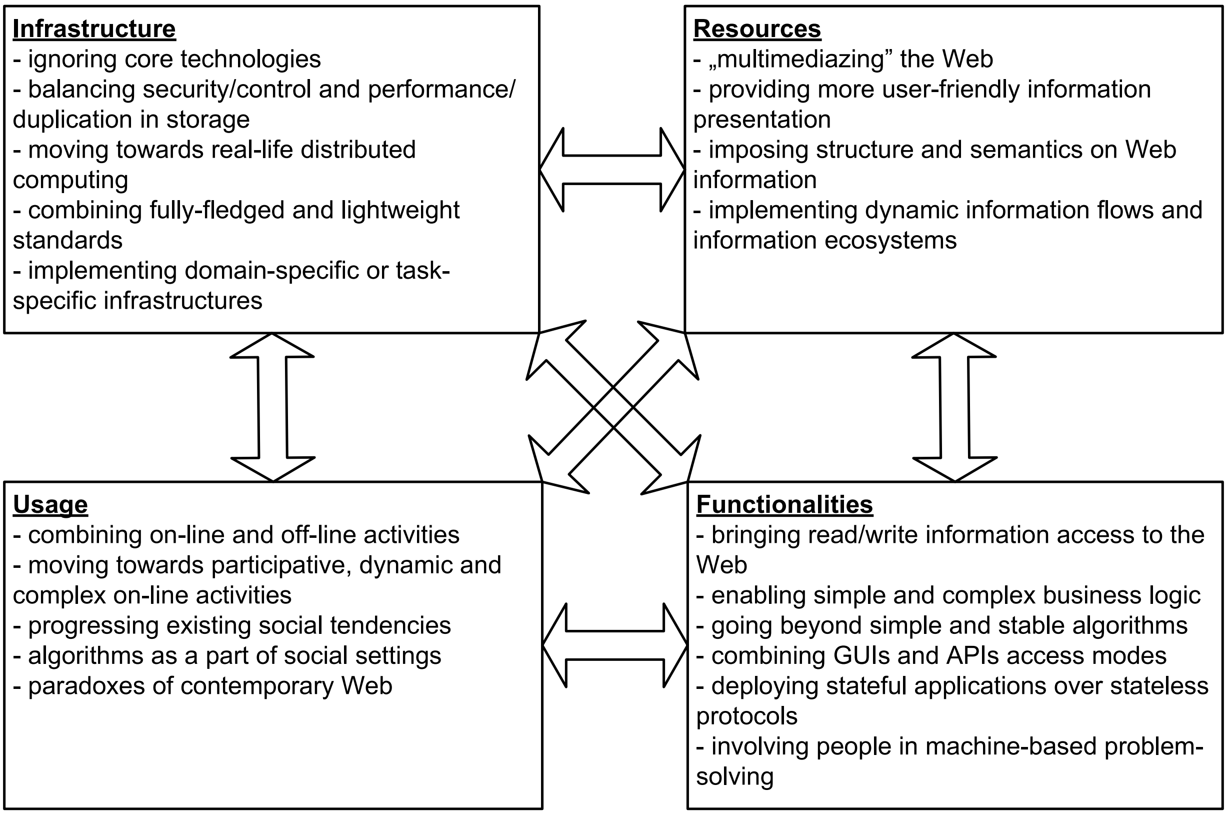

Infrastructure
- ignoring core technologies
- balancing security/control and performance/ duplication in storage
- moving towards real-life distributed computing
- combining fully-fledged and lightweight standards
- implementing domain-specific or task-specific infrastructures
Resources
- „multimediazing" the Web
- providing more user-friendly information presentation
- imposing structure and semantics on Web information
- implementing dynamic information flows and information ecosystems
Usage
- combining on-line and off-line activities
- moving towards participative, dynamic and complex on-line activities
- progressing existing social tendencies
- algorithms as a part of social settings
- paradoxes of contemporary Web
Functionalities
- bringing read/write information access to the Web
- enabling simple and complex business logic
- going beyond simple and stable algorithms
- combining GUIs and APIs access modes
- deploying stateful applications over stateless protocols
- involving people in machine-based problem-solving

Fig. 1.1. *Four components of evolving Web*

2.1. Three Areas of Web Infrastructure. On-line infrastructure includes three main areas: standards, storage and computing. Standards propel all kind of communication and exchange on-line, thus they are the prerequisite for effective data flow, applications integration and business processes execution. In recent years we observe quick development of standards geared towards interoperability of data and distributed software.

Apart from standards developed or supported by standardization bodies (such as SOAP, RDF, OWL and OpenID), a number of formats and interoperability protocols (such as microformats, JSON or RESTful services) became *de facto* standards thanks to their wide adoption (see Table 2.1 for examples). They often solve the same problems as official standards in a less complete and flexible, but also simpler and easier to implement way. While the friction between competing standards causes confusion and bears new implementation challenges, it also results in better choice for developers and quicker maturing of new technologies.

Table 2.1
Fully-fledged standards and their lightweight counterparts

Area	**Fully-fledged standards**	**Lightweight (de facto) standards**
Remote calls	SOAP, CORBA, RMI	RESTful services, XML-RPC
Structural representation	XML	JSON, (X)HTML
Semantic representation	RDF, OWL, WSML	microformats in HTML
Federated identity	OpenID	e-mail as login
Metadata	Dublin Core	folksonomies
Portlets/Gadgets	JSR 286	Google Gadgets

One of the foundations of convergence of Web solutions, that we also perceive as a pseudo-standardization process, is the cultural tendency towards reusing best practices of other users and businesses. This results in similarities between business processes of many on-line businesses, multiple sites sharing similar information organization schemas and even textual documents of specific types (e.g. advertisements or calls for papers) sharing their structure, formatting and layout features. It is worth noting that there are counter forces preventing total unification—they are driven by need for competition and differentiation of information both in terms of its content and processing capabilities, which are partially dependent on standards for information sharing. Thus we observe interesting process that throughout the years pushed the limit of standardization: first application protocols were agreed upon, later data representation formats were converging (this process is finalized currently), with final step in standardization of languages enabling flexible extensions to information representation formats, enabling both standardized processing and flexibility that enables value-added processing.

The second area of basic Web infrastructure consists of content and data storage facilities. It is shaped by two conflicting requirements, depicted in Figure 2.1. The former is to have maximal control over information location and its access rights—promoting storage centralization and forcing self-management (together with lack of affordable services to outsource storage). The latter requirement is to assure maximal performance (i. e. short access time from multiple locations, as well as storage scalability and persistence) and cost effectiveness—which is promoting distributed storage and outsourcing of the storage facilities.

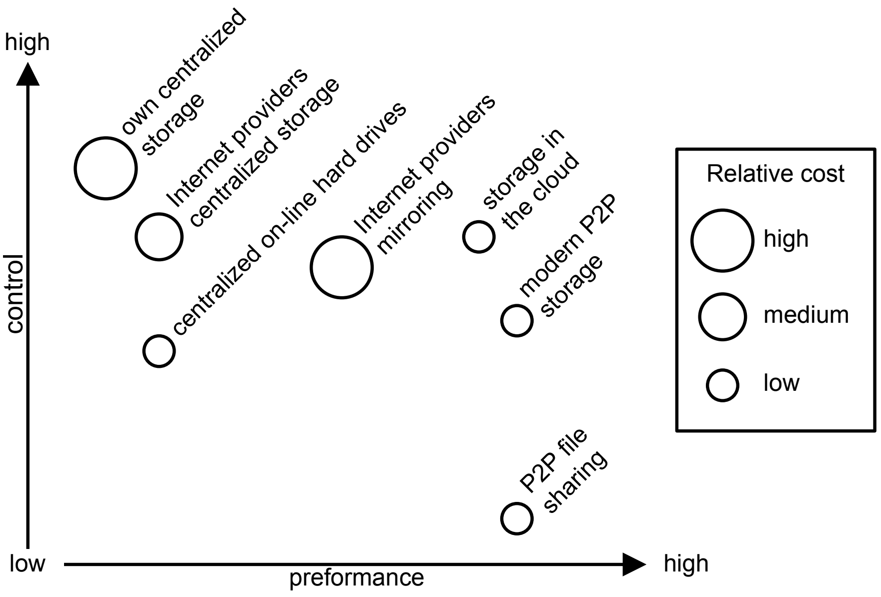


Fig. 2.1. *Control/performance tradeoff in storage solutions*

A few years ago the storage options were scarce: unless one was Yahoo! or Google, (s)he could only maintain own Web servers (typically more expensive and not necessarily more secure solution), or use individual servers made available by Internet providers (typically less expensive but somehow limited in functionality). In both cases, mirroring and using broadband connections were virtually the single options of increasing performance.

The first change we have witnesses was popularization of peer-to-peer (P2P) file-sharing applications. Peer-to-peer file storage proved scalable and assured rather good persistence of content (often against the will of its original creators or owners). However, in its pure form it also meant extreme lack of control over location and flow of information, making it absolutely inapplicable in business or personal information management scenarios. It is only after few adaptations, that P2P protocols found their way to the Web, making high-performance low-cost streaming of multimedia content more feasible (with BitTorrent being one of icons of this transformation). Today's distributed, cloud-based databases (such as Google's BigTable [7] or Amazon's SimpleDB[1] and Simple Storage Service[2]) learned lesson from both typical hosting and peer-to-peer systems, proposing what seems to be a good control-performance trade-off. Similarly as in case of hosting, the content is taken care of by a single

[1] aws.amazon.com/simpledb/.

[2] http://aws.amazon.com/s3/.

company. Similarly as in case of peer-to-peer systems information is distributed and replicated in multiple locations all over the world. However, in contrast to P2P networks, cloud-based storage is geographically stable and closed, thus does not suffer from high churn of nodes. Thanks to economies of scale, the proposed solutions are at least as affordable as hosting, with better performance, almost perfect scalability and usage-based cost calculation. In majority of cases distributed storage has higher uptime, even if spectacular failures happen (and may have high impact at least at the psychological level).[3] These failures encourage others (e.g. P2P storage Wuala[4]) to look for other solutions with a little bit more of a twist towards performance at expenses of control. Thus, cloud-based distributed storage surely is not the final answer to the centralized vs. distributed storage conflict.

The third area of Web infrastructure is related to on-line computing. Similarly as in case of storage, a few years ago this part of infrastructure was dominated by private or hosting-based servers using more or less standardized configurations (e.g. LAMP/WAMP[5] or Java-based technologies) to enable easy deployment of typical software solutions. Since then important changes occurred, leading to development of the rich computing environment that the Web is today. First revolutionary change is related to public accessibility of cloud computing platforms available form a number of companies including such huge players as Amazon (Amazon EC2[6]), Google (Google App Engine[7]) and Microsoft (Azure Services Platform[8]). These solutions, roughly classified as "platform as a service" (PaaS) solutions (e.g. Google App Engine, force.com) and "infrastructure as a service" (IaaS) solutions (e.g. Amazon EC2), make Web applications more scalable and available from all around the world. Moreover, as cloud computing platforms charge on per-usage basis, they are affordable for everyone and more economically reliable than previous solutions. Although spectacular failures of clouds generate a lot of fuzz, their uptime remains higher than for typical hosting solutions. In parallel, a shift towards virtualization enabled to build custom application stacks, and run them on multiple servers, or on cloud infrastructure (e.g. Amazon's EC2). Thus, today it is much easier to set up non-standard, scalable, high-performance servers, required by many specific Web-based services.[9]

2.2. Domain-Specific Infrastructures. Another rapid change at the edge of infrastructure is the development of domain-specific platforms that enable to build instances of specific applications with little effort. For examples ning[10] enables easy creation of social networking sites, Facebook Platform[11] enables development of applications using Facebook features and users base and Yahoo! BOSS[12] supports creation of custom search engines (and promises sharing revenue soon). A number of platforms for development of e-stores (including Yahoo! Store[13] and eBay Stores[14]) exist (the extreme example is Zlio.com[15] - in this case shop owner's activities are limited just to building a Web site and choosing product range; ordering, payment and logistics are supported by Zlio itself). Other examples include services such as TinyURL[16], Bit.ly[17] and purl[18] that aim at becoming another layer of standardized resources addressing on top of DNS. Another areas where some players aspire to become default infrastructure include enactment of complex information flows (Yahoo! Pipes[19] is the most renown example of such service), automated translation services (with tools such Google Translate[20] and Yahoo! BabelFish[21] competing with many smaller businesses), contextual ads (area strongly dominated

[3] See: `http://www.readwriteweb.com/archives/google_failures_serious_time_t.php`.

[4] `http://www.wuala.com/`.

[5] Linux/Windows + Apache + MySQL + PHP.

[6] Amazon Elastic Compute Cloud, `http://aws.amazon.com/ec2/`.

[7] `http://code.google.com/appengine/`.

[8] `http://www.microsoft.com/azure/`.

[9] Design of custom application stacks for virtual servers and cloud computing is simplified by services such as Elastic Server on Demand, `http://elasticserver.com/`.

[10] `http://www.ning.com/`.

[11] `http://developers.facebook.com/`.

[12] `http://developer.yahoo.com/search/boss/`.

[13] `http://smallbusiness.yahoo.com/ecommerce/`.

[14] `http://stores.ebay.com/`.

[15] `http://www.zlio.com/`

[16] `http://www.tinyurl.com`.

[17] `http://bit.ly/`.

[18] `http://purl.org/`.

[19] `http://pipes.yahoo.com/`.

[20] `http://translate.google.com/`.

[21] Originally developed for Altavista, now available at `http://babelfish.yahoo.com/`.

by Google AdWords[22]), on-line conference management (with EasyChair[23] being probably the dominant player and support for social network and content portability (with Gnip[24] being top commercial example, and SIOC community being the research leader [5]). It is also to be noted that a number of specific APIs were created with the objective of becoming standard infrastructure in specific applications areas. Examples include OpenCalais[25] from Reuters for natural language processing, Fire Eagle[26] for storing and manipulating location data, Mozilla Weave[27] - for storing and sharing data on browsing sessions, bookmarks etc. Finally, few infrastructure-like APIs focus on involving people into problem solving in multiple complex areas, such as information extraction, organization, integration and cleansing. This involvement takes multiple forms, including explicit (and paid for) people actions (as in case of Amazon Mechanical Turk service[28] or other forms of crowdsourcing different business activities including content creation, problem solving and even R&D [17]), and using analysis of behaviors of large groups of Internet users (for example in user reviews mining [18]).

3. Resources. The growth of size of resources available on-line has two faces: on one hand, we observe quick growth of quantity of content (i. e. unstructured information both in textual and multimedia form), on the other hand the Web is also the biggest repository of data (i. e. structured and semi-structured information). In both cases the changes are not only quantitative but also qualitative: the way data and content is made available on-line is evolving rapidly towards two (often opposed) objectives: one is representation better adjusted to needs of users and other is the form easily processable by machines. Example of these two tendencies are represented in Figure 3.1 and discussed in two following sections.

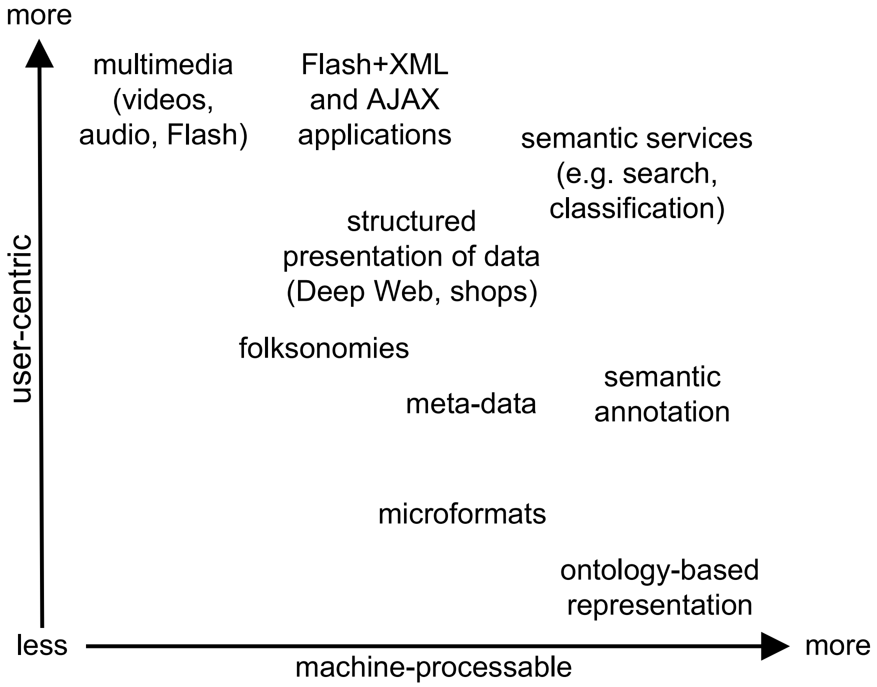

more
multimedia (videos, audio, Flash)
Flash+XML and AJAX applications
semantic services (e.g. search, classification)
structured presentation of data (Deep Web, shops)
user-centric
folksonomies
meta-data
semantic annotation
microformats
ontology-based representation
less
machine-processable
more

Fig. 3.1. *User-centric and machine-processability-oriented tendencies in functionalities*

3.1. User-centric tendencies. The tendency of making content and data more adapted to human users takes two main angles: on one side it affects the content and data themselves, on the other side it influences how the content is presented. Firstly, the online content in recent years has become more multimedia and visually

[22] http://adwords.google.com/.
[23] http://www.easychair.org.
[24] http://www.gnipcentral.com/.
[25] http://www.opencalais.com/.
[26] http://fireeagle.yahoo.net/.
[27] http://labs.mozilla.com/projects/weave/.
[28] https://www.mturk.com/.

appealing: thanks to wider broadband access, audio and video content become accessible to the vast part of Internet users. As a result more information previously provided in text form took much richer presentation: for example, growing part of software producers provides instructive videos apart from (or even instead of) text manuals, and more and more news are provided to users using podcasts. Secondly, thanks to such technologies as dHTML, AJAX[29], Adobe Flash and Silverlight, many on-line resources are not only multimedia, but also interactive and non-linear. These possibilities are for example widely used in different kinds of on-line training. It is to be noted that these changes often happen at the expense of accessibility, readability, and "skimability" of provided information, especially for people with special needs [8, 26].

The evolution of content presentation is mostly related to advance of dynamic user interfaces that try to mimic desktop software interaction paradigms (e.g. drag and drop or complex controls) and response times (e.g. by avoiding reload of the whole page on link click, using AJAX). Such rich user interfaces become a medium of its own, not easily separable from the content [24]. In extreme situations content is not a static, stable entity at all, and is recreated each time by a sequence of operations (happening both client and server-side) controlled by user actions, usage context (e.g. time, location of user) and external factors (e.g. other user's actions, real-life phenomena, random elements generated by algorithms). As an example - such dynamic content and its presentation is typical for real-time search engines (e.g. Twitter search) or highly personalized search facilities.

Similar tendency at the presentation layer is happening in case of some data-intensive Web sites. For example Flash or AJAX technologies are often used for interactive data selection or on-demand download of more details for already displayed data. In some cases, similar technologies are also used for visualization of data (not provided any more in textual form), for example as charts (in case of numerical data[30]) and simulations or models (e.g. in case of body colors in car industry).

Data and content presented using rich user interfaces are often called "dynamic". However, one more dimension of content and data dynamism should be also considered. Following the paradigm of collaboratively developed and maintained content, the growing amounts of information on-line are always-non-final, continuously evolving resources. This tendency touches even such traditionally stable entities as books (Wikibooks) or journal articles (scientific blogs). In parallel, mechanisms of partial control such as versioning and branching become popular. Similar dynamism can be observed in case of data; in this case quick changes result from tight connection to dynamic processes (e.g. in case of price lists, popularity of news articles or search result) or to measurement of dynamically evolving external conditions (e.g. sensor-based weather analysis). As a result, more and more content and data objects should be interpreted more as streams of new information (as in case of blogs, Twitter messages or sensor-based data sources) or of information updates (as in case of price lists and Wikipedia revisions), rather than stable entities.

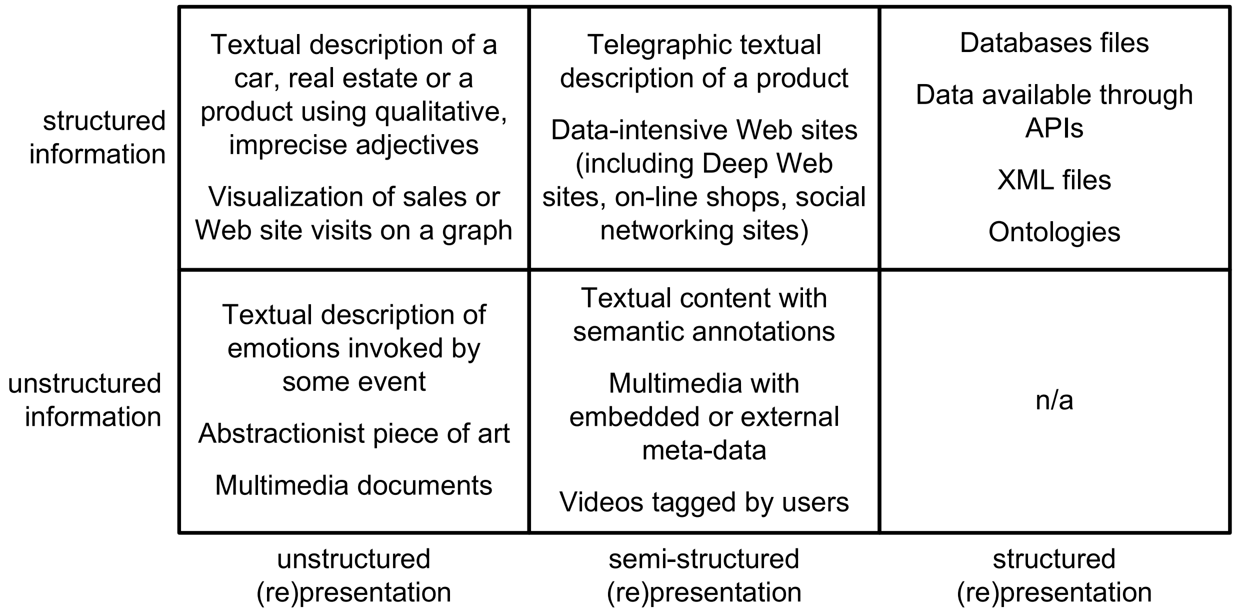
3.2. Machine-processability tendencies. In parallel to the evolution of content and data format and presentation, we observe quick changes related to machine processability of information.

Firstly, a number of structure-centric, semantics-aware formats were proposed (as mentioned previously in the context of standard infrastructures). They may be used both to store metadata of on-line resources (e.g. title, categories, creator or tags of specific document), and to incorporate inline annotations into on-line documents (e.g. concerning specific named entities, numeric values or key phrases). Examples of used formats include family of XML technologies (XPath, XQuery, XSLT), RSS, RDF and OWL. Used meta-data and annotation schemes include microformats, Dublic Core, domain-specific ontologies such as FOAF and SIOC, MPEG-7 standard (for multimedia), and a number of non-standard annotation schemes proposed by different services.

Secondly, machine-processability of the content grow thanks to two complementary strategies to provide structure and semantics of data and content: bottom-up approach and top-down approach. In bottom-up approach the structure and semantics are imposed on content the by authors or Internet users by modification of underlying technology, or manual enrichment of the content. In most cases, this approach provides good quality structural and semantic information embedded directly in the content, typically keeping its human-readable character. While bottom-up approach is an important research topic and the number of sites that give some support to this approach is growing, the adoption of structured and semantic representation is still low, due

[29] While this term standard for asynchronous JavaScript and XML, it is also often used for asynchronous update of pages by using formats other than XML, such as JSON, XHTML or proprietary formats.

[30] See `http://www.tate.org.uk/netart/bvs/thedumpster.htm` for an interesting example for blogs visualization or Google Analytics motion charts



	unstructured (re)presentation	semi-structured (re)presentation	structured (re)presentation
structured information	Textual description of a car, real estate or a product using qualitative, imprecise adjectives Visualization of sales or Web site visits on a graph	Telegraphic textual description of a product Data-intensive Web sites (including Deep Web sites, on-line shops, social networking sites)	Databases files Data available through APIs XML files Ontologies
unstructured information	Textual description of emotions invoked by some event Abstractionist piece of art Multimedia documents	Textual content with semantic annotations Multimedia with embedded or external meta-data Videos tagged by users	n/a

Fig. 3.2. *Levels of structure of information and presentation*

to weak incentives, relatively high costs and missing standardized vocabularies. All of these reasons propel the development of top-down approach, based on automated processing of Web content. In this approach the hints already present in the content are used together with external resources in order to structure and "semantify" information, without direct co-operation of individual Web sites. The output of automated top-down processing is more "digestible" (e.g. better structured, aggregated, organized or summarized) to users or machines than original content. This approach combines different techniques of Web content mining (such as classification and clustering, text summarization, information extraction, relations mining, ontology learning and population, opinion mining or multimedia content analysis), Web structure mining (such as measuring importance of Web sites, community discovery based on dense subgraphs of Web graph, Web site complexity measurement and Web pages categorization) and Web usage mining (such as discovery of customer clusters, analysis of products or documents popularity, improvement of collaborative filtering). [20, 23, 6] Search engines are an classical example of top-down approach—by using some general mining rules, they impose a specific ordering (by some measure of relevance to keywords and analysis of link graphs), specific structure (snippets reflecting contents of given page) and similarity-driven mechanisms (such as search for similar pages or clustering of results). First search engines used only Web content mining techniques. Then we observed Google's break-through PageRank algorithm using Web structure mining. Today major search engines use also to some extent behavioral analysis based on Web usage mining. On the other hand, a few smaller and ambitious players, such as Hakia[31], PowerSet[32] (recently acquired by Microsoft) and Evri[33], aim at enriching content not only with cited types of structure, but also with semantics. Some of today approaches to structuring Web content share properties of both bottom-up and top-down methods. Examples include social tagging and bookmark management sites such as del.icio.us[34], sites calculating other Web sites popularity based on votes such as social news site digg[35] or PostRank[36]—blog posts assessment service, different kinds of content annotation services such as SpinSpotter (allowing to annotate non-objective passages in newspaper articles)[37], and sites restructuring scripts or applications such as Dapper[38]. On one hand they are similar to top-down approach, because the structurization is happening outside of Web

[31] http://www.hakia.com.

[32] http://www.powerset.com.

[33] http://www.evri.com/.

[34] http://delicious.com

[35] http://www.digg.com/.

[36] http://www.postrank.com.

[37] See: http://www.spinspotter.com/.

[38] http://www.dapper.net.

site whose content is being structured, and because their approaches are general, often domain-independent, possibly large-scale and typically based on specialized algorithms. On the other hand, similarly to bottom-up approaches they are based on manual work rather than fully automated.

Thirdly, important contribution to machine processability comes from methods that enable both assessment of identity of multiple objects and measurement of their similarity are developed. Thus, both data and content objects are more and more connected and related to other entities. This area is strictly related to well-known research fields of schema mapping and matching of individual records or instances (which are a part of a number of information management tasks such as data integration, data cleansing and ontology merging). Moreover, quick progress in this areas influences both bottom-up and top-down solutions. Top-down solutions in schema mapping concern continuous improvements in methods of automated schema mapping. In recent years this area evolves towards holistic approaches, that enable mapping of multiple (often meaning: very large number of) schemas at once [14]. Related concept of "dataspaces" [13] seems to be implemented in real-life by Google Base, that gathered over 100K schemas and should allow large-scale schema mapping. A lot of top-down methods at the instance level were also proposed, using both more elaborate similarity measurement functions and better lexical resources. Significant body of research into ontology mapping and merging also fits to a large extent to this philosophy. On the other hand, ontologies are also the representation that promotes interconnection of multiple knowledge bases both at the class and instance level (linked data philosophy[39]). In bottom-up approach schema and ontology mappings and records equality are defined manually or by specific transformation software (varying from stand-alone procedural tools to declarative queries of rules executed by specific engines). Domain specific, dictionary-based records linkage is for example typical for shopping bots that help compare prices of the same products in different locations. Another helpful bottom-up tendency concerning instances is related to standardization of object properties formats (e.g. XBRL has been recently accepted by U.S. Securities and Exchange Commission as the required format for financial reports of public and mutual fund companies[40]) or to popularization of domain-specific identifiers (such as DOI[41] for electronic documents, or OpenID[42] for people). Bottom-up and top-down changes to the Web are happening simultaneously, and support one another. Even limited range of structure added to Web content may significantly lower the difficulty of top-down tasks. For example, usage of additional information encoded in user tags proved to be useful for Web content summarization [25], and potentially can have positive impact on performance of Web search [16]. Intuitively, when microformats are used, the task of information extraction (as well as tasks that depend on it, such as analysis of on-line social networks) should become much more feasible. Similarly, usage of tags may simplify the task of record linkage. On the other hand, top-down approach may significantly reduce costs of creation of semantic representation of content. It may even fully automate this process in some domains.

3.3. Content Flow and Content Ecosystems. One of characteristics of on-line data and content is their dynamic flow between a number of services. Originally posted to a single Web site (e.g. blog, shop price list or on-line database) or discussion list, the information may be reposted in a number of forms in other locations. Similarly, the changes to original content may be further propagated to a number of other locations. Complexity of such flows in case of blog posts is demonstrated by Figure 3.3.

The propagation of the content on the Web can be done by pop and push information flows. The former are initiated by the service that acquires a copy, and the latter is activated by information author or the service that the content is originally posted to. Examples of pop information flows include indexing by search engines or synchronization through RSS, examples of push flows include mirroring of content or submission of the same information to multiple Web sites.

In the same time the flows may be manual (fully performed by people), semi-automatic (requiring some setup activities but afterwards performed automatically) or fully automatic (requiring no user interaction at all). Examples of manual flows include quoting or copying content to other locations or forwarding it to friends. Examples of semi-automatic flows include mashups created with Yahoo! Pipes or YouTube videos embedded in a blog post. Typical examples of automatic flows are related to indexing and caching by search engines, or to usage of user comments on products for their automated qualitative assessment.

[39] http://linkeddata.org/.

[40] See: http://www.google.com/hostednews/ap/article/ALeqM5jTRoSiNGE5B07igsMWNH3Z0tbmAQD954M4800.

[41] http://www.doi.org.

[42] http://openid.net/.

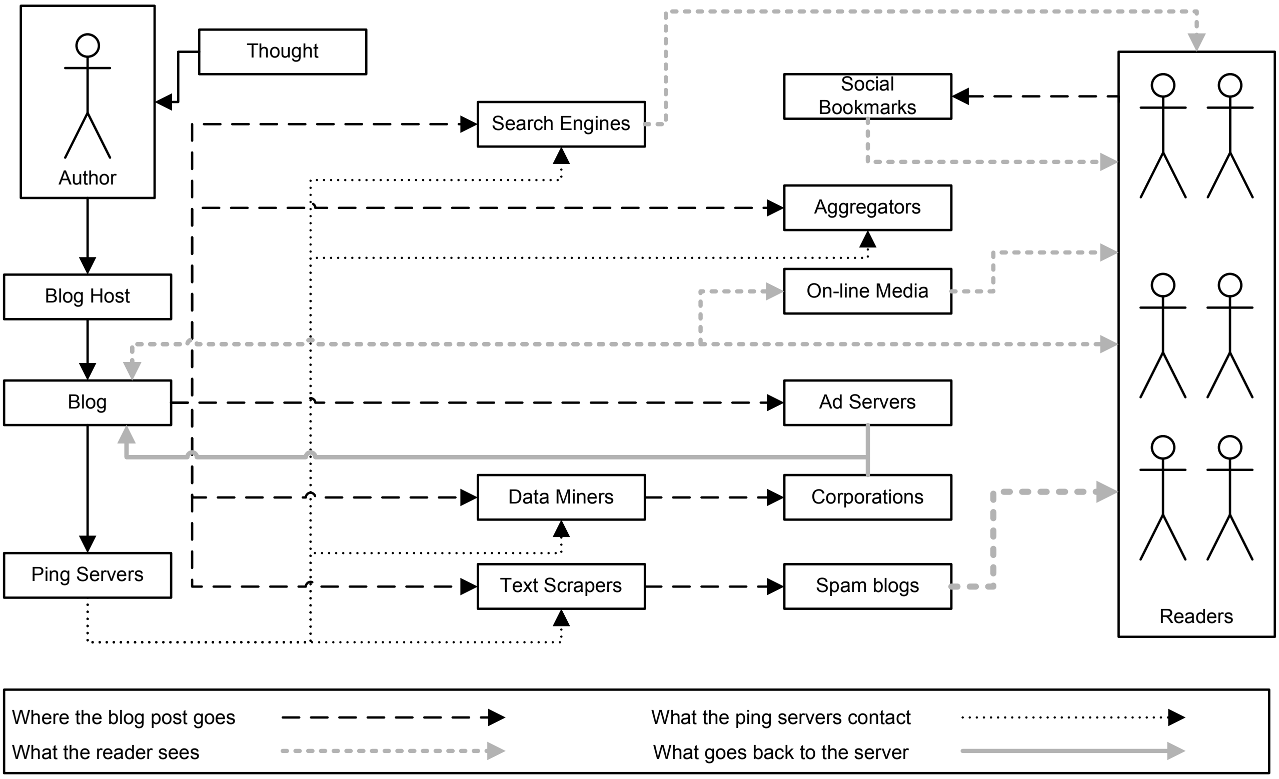


FIG. 3.3. *The life cycle of blog post as an example of complex flows (based on [27])*

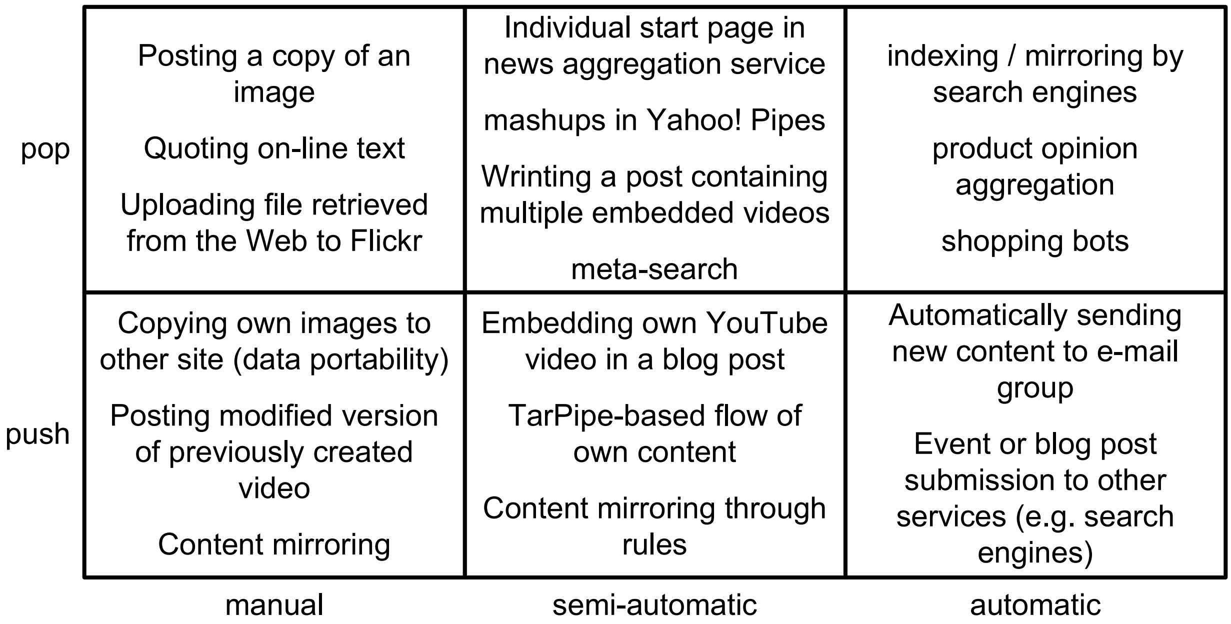
Finally, the flows may preserve the identity of original content (e.g. in case of mirroring or embedding of content) or can do some transformations on content (e.g. adding semantics or hyperlinks, summarizing multiple user comments, or quoting a fragment of original text). For example automatic repost of e-mail group messages to Web-based archives or embedding of videos preserve the identity of original content. Machine translation services aim at providing the same content in different languages. Some NLP-based services (examples include eventSeer[43] in the area of calls for conference papers, or a plethora of services using OpenCalais) automatically add new links or meta-data. Finally, shopping bots, summarizing services (such as semantalyzR[44]) and other services using information extraction reuse only small portion of original content.

Different types of content flows are compared in Figure 3.4. It is also to be noted, that apart from flows that result in creation of new (instances of) content, information may be also accessible outside its original location via different types of querying services including meta-search engines, on-the-fly search tools (e.g. Twitter search[45]) or on-demand translations services (that do not store translated texts).

It will be interesting to see what will be the impact of content flows and ecosystems to information transparency. It seems that currently there is little care given by content aggregators and services that transform it to providing information about the original source (and additional metadata such as time of retrieval) of information and actual transformations performed. As the phenomenon matures and is wider applied in business scenarios we might observe new formats emerging to provide such metadata and forces driving to increase transparency of processing.

4. Functionalities. The area of functionalities is concerned with all kinds of operations that can be performed on on-line resources. This area has undergone major changes from the beginning of Internet era. It is visible even in the case of basic operations related to access to data or content. They concern both the "read" access to Web resources (that was possible from the beginning of WWW), and different types of "write"

[43] `http://www.eventseer.net/`.
[44] `http://semantalyzr.com/`.
[45] `http://search.twitter.com/`.



	manual	semi-automatic	automatic
pop	Posting a copy of an image Quoting on-line text Uploading file retrieved from the Web to Flickr	Individual start page in news aggregation service mashups in Yahoo! Pipes Wrinting a post containing multiple embedded videos meta-search	indexing / mirroring by search engines product opinion aggregation shopping bots
push	Copying own images to other site (data portability) Posting modified version of previously created video Content mirroring	Embedding own YouTube video in a blog post TarPipe-based flow of own content Content mirroring through rules	Automatically sending new content to e-mail group Event or blog post submission to other services (e.g. search engines)

Fig. 3.4. *Examples of different types of information flows*

access (that were originally rarely supported on the Web, although foresaw by creators of the basic Internet infrastructure).

"Read" access to information has been transformed mainly thanks to already discussed bottom-up and top-down structurization and semantification of Web content and data. "Write" access to content and data includes relatively recently popularized features such as possibility of content editing (e.g. in case of wikis), creation of new content (also by duplication and edition of existing content), extension of existing content (e.g. by tagging resources, adding comments to articles or forum posts, or writing additional statements as in case of microblogging) and addition of new data, influencing aggregated data quality (e.g. voting in rankings, "digging" content, providing feedback on visited sights or hotels, or providing information on weather conditions[46]).

4.1. Business Logic Functionalities. Apart from such basic, storage-related functionalities, the on-line services implement different kind of business logic that focus on solving specific problems based on external or internal information. They include complex operations such as transformation, discovery, analysis, comparison, search and ranking of different types of information. The logic itself may have very different construction. It may be based on a stable algorithm (e.g. conversion of different measurement units, basic tax calculation), parameterized algorithm (e.g. tasks involving currency conversion, or tax calculation with changing tax rates or list of exempted products), algorithm applying user-provided rules (e.g. on-line content filtering based on preferences specialized by an user), machine learning algorithms (e.g. spam filtering functionalities), and interactive algorithms requiring participation of user or querying of external knowledge sources (e.g. search for ambiguous locations with search engines or map services). While majority of logic components provide exactly one and final resultset for specific input parameters, in some cases the logic may iteratively provide series of improved result sets (e.g. calculated with more iteration of optimization algorithms or constructed based on larger set of input data), based on a kind of subscription to results of on-demand calculation (like constant reordering of search results in some meta-search engines based on new data coming from multiple indexes).

While Web protocols are constructed as stateless, both stateless and stateful applications can be constructed on top of them. In case of stateless applications, activities (or invoked procedures) have no impact on results of future activities (or invocations) of the same user nor of other users. In stateful application current activities have impact on result of future activities with the same session (with the state stored temporarily), or also between sessions (with the state stored in a permanent way). The stored state can be itself meaningful to the

[46] For example in case of OtherWeather.com.

user (e.g. different aspects of manually typed user profile) or contain values that are solely machine-interpretable (e.g. vector representation of user interests based on keywords (s)he entered). Finally, the state may be attached to specific user (when login identification is used), to specific IP address, specific Web browser (using Cookies), some combination of the above, or may be shared by a number of users (e.g. the list of available tickets in on-line ticket sale service).

	personal	shared
meaningful	shopping cart on-line playlist chosen search criteria list of contacts in social networking sites or instant messaging	free seats in on-line reservation sites number of goods at stock collected money (fundrising for electoral campaign or charity) Wikipedia articles
meaningless	aggregated browsing history internal state of an on-line arcade game	browsing patterns collaborative filtering statistics internal state of a multiplayer game

Fig. 4.1. *Examples of meaningful / meaningless and personal / shared state*

Figure 4.1 provides a number of examples for different types of state of business logic components.

4.2. Access Modes to On-line Functionalities. On today's Web, different functionalities are accessible in two basic modes: through Web-based GUIs and by different kinds of APIs. The first mode is focused on providing the access to features of on-line applications to human users. In this approach, Web operations are typically invoked by user entering specific pages, filling in forms or performing other HTML-based activities in Web browsers (such as clicking or dragging objects). However, some of these activities may use specific, non-Web technologies (such as Flash, Java or Sliverlight). While majority of logic in Web sites and Web applications is executed on server-side, more and more features are fully client-side. In many cases client logic is used as "a glue" combining functionalities provided by other server-side services (e.g. in case of mashups, widgets and embeddable JavaScript libraries such as Web analytics trackers). However, in some cases it may be accessible even purely in off-line mode (as in case of Google Gears[47]), blending the distinction between Web applications and desktop software. This blending goes even further with different types of business logic pluggable in user's browser with methods varying from lightweight (such as bookmarklets), through plugins using basically the same Web technologies but with greater access rights (e.g. FireFox plugins, Opera widgets, some Java and Flash applications), to fully integrated binary extensions such as Internet Explorer toolbars. At the extreme we can find desktop applications that embed Web browsers (e.g. for visualization or content access purposes) but have their logic hardcoded.

The second access mode, based on different type of APIs, is related to usage of on-line services by other software components. API types vary from complex and standardized (SOAP-based Web Services), through lightweight but mostly standardized (XML-RPC) to lightweight and mostly unstandardized (many REST-ful services with more or less stable and formalized response formats). It enables any applications to easily access and compose pieces of logic provided by multiple on-line services, as well as to access multiple types of on-line resources. Nowadays, this composition can be a part of client-side business logic of specific GUI-centric Web application (i.e. can be used internally by specific Web sites), it can be performed in a form of mashups

[47] `http://gears.google.com/`.

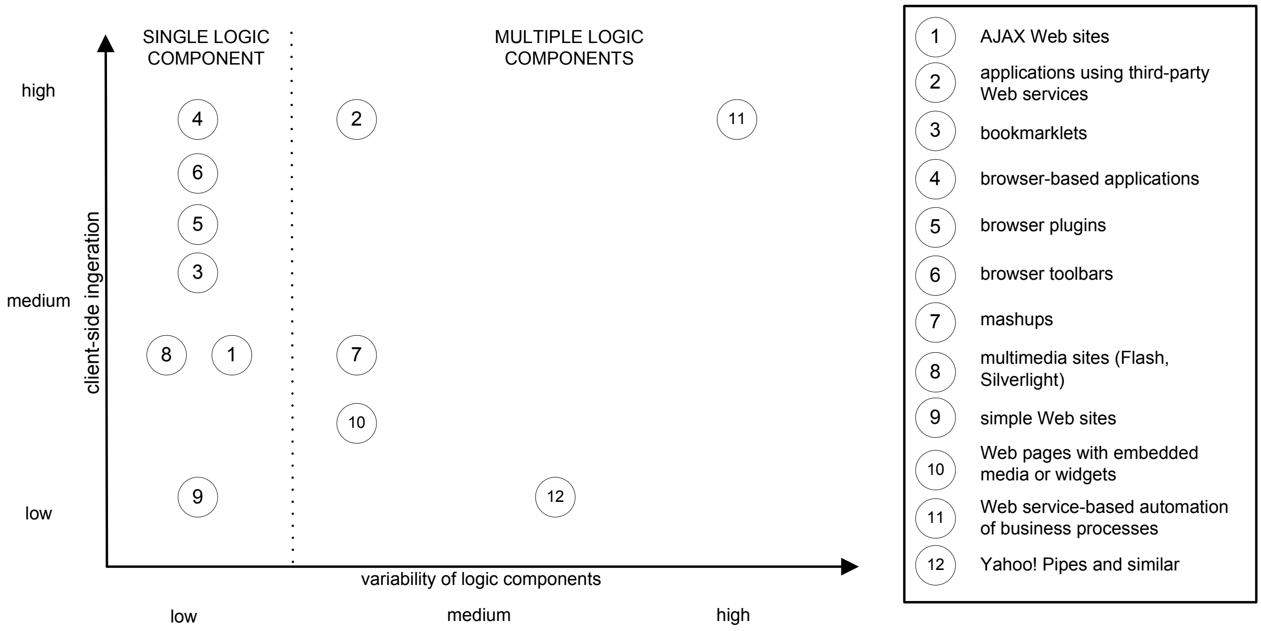

SINGLE LOGIC COMPONENT
MULTIPLE LOGIC COMPONENTS
high
medium
low
client-side ingeration
variability of logic components
low
medium
high
4
6
5
3
8
1
9
2
7
10
12
11
1 AJAX Web sites
2 applications using third-party Web services
3 bookmarklets
4 browser-based applications
5 browser plugins
6 browser toolbars
7 mashups
8 multimedia sites (Flash, Silverlight)
9 simple Web sites
10 Web pages with embedded media or widgets
11 Web service-based automation of business processes
12 Yahoo! Pipes and similar

Fig. 4.2. *Different ways of implementing and exposing functionalities*

developed by programmers, or constructed using visual mashup construction tools (such as Yahoo Pipes). A lot of research also focuses on using semantics for composition based on technically underspecified business processes or objectives to be attained. It is to be noted that automated access to Web site content or functionalities may be also enabled if no API is provided, by using Web data extraction ("screen scraping") and navigation automation tools (such as Dapper, GreaseMonkey[48], WebVCR [2], iMacros[49]).

4.3. Involving People in Complex Functionalities. Typically when we think about business logic we mean automatically performed activities based on some pre-defined rules or algorithms. However, the open character of both Web and Web-based APIs makes the business logic potentially (co-)executed by people (individuals, businesses or groups of people). Human participation in composite logic may be synchronous (taking part at the moment of logic execution) or asynchronous (happening later). It may be also direct (with user actively taking decisions and actions) or indirect (with decisions being byproduct of other user activities, possibly aggregated over time by use of different machine learning methods). Few examples of direct and indirect, synchronous and asynchronous involvement of people in complex functionalities are gathered in Figure 4.3.

4.4. Examples of Typical On-line Functionalities. Typical on-line services focus on offering a few basic features. They include:

- information access and search—acquisition of information from Web sources,
- information management—management of document and media, including authoring, modifying, sharing, versioning, downloading,
- information transformation—transformation of one kind of information into different one,
- communication—spoken or written free-text exchange of information between people,
- collaboration and problem-solving—support for solving of complex problems by community,
- entertainment—individual or social hobbies, games etc.,
- self-development—education, training and spiritual development,
- business and transactions—acquisition and sale of goods and services on-line.

Apart from supported features two other dimensions may be used to classify Web sites. They are characteristics of the medium and properties of the application itself. Some of the most important properties of medium

[48] http://www.greasespot.net/.
[49] http://imacros.net/.

	direct	indirect
synchronous	(Some) Amazon's Mechanical Turk services Services that require interactive user decisions (e.g. seat selection, selection of specific flight) Services that require user to expand or disambiguate query (e.g. some search-based services	Discovery of "hot news" topics based on Twitter search Suggesting media (videos, audio files) based on what your contacts watch / listen to at the moment
asynchronous	(Some) Amazon's Mechanical Turk services Correction of automatically translated texts (post-edition) Removal of automatic annotation or automatically added links Correction of search results	Improving product similarity measures with data on sets of items previously compared by users Including statistics on results clicked by other users in relevance calculation functions Collaborative adaptive Web sites

Fig. 4.3. *Examples of different ways of involving people in complex functionalities*

are: rhythm (synchronous / asynchronous), bandwidth consumption (low / high), format (text-based / voice) and permanence (persistent / ephemeral) [11].

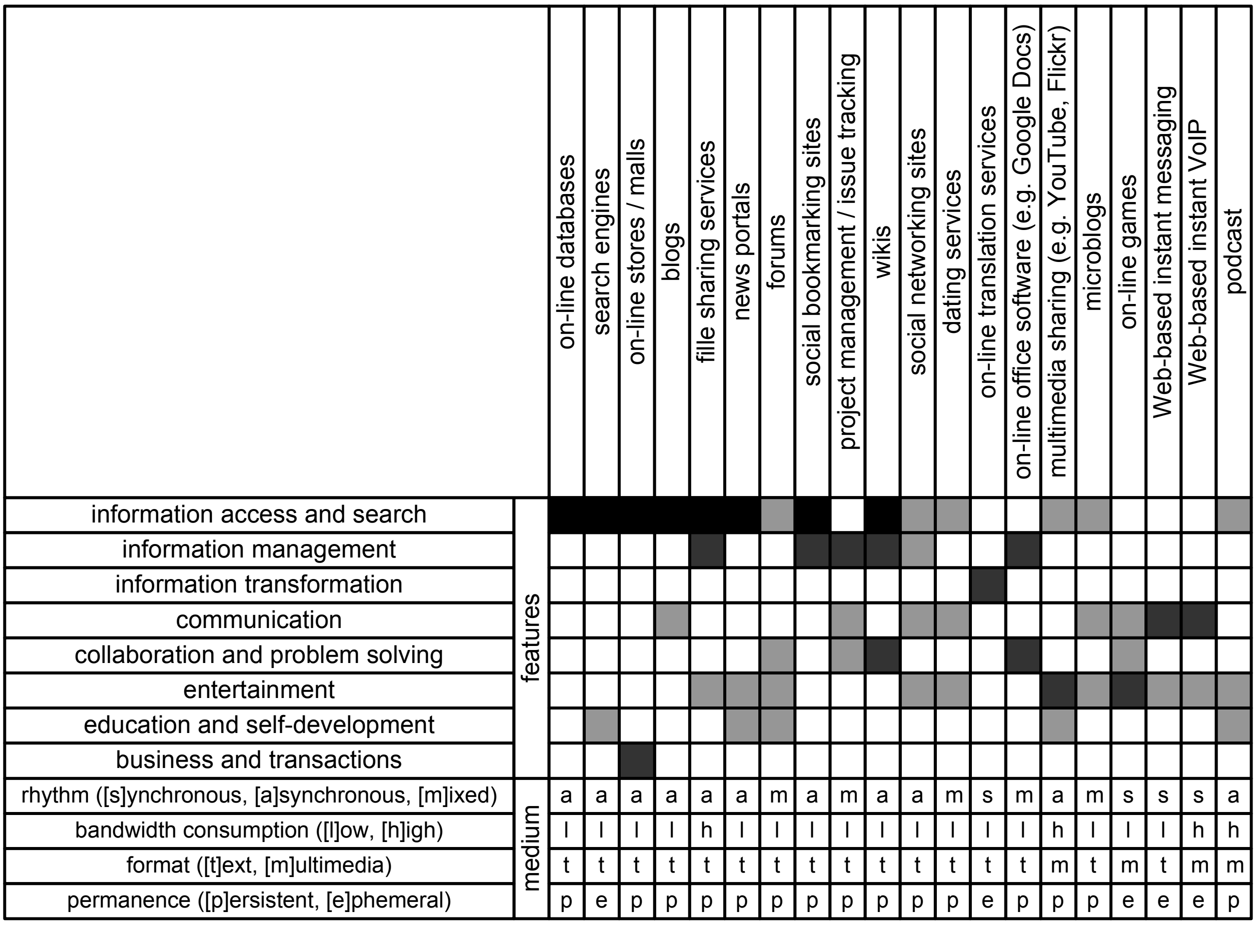
Figure 4.4 compares a number of on-line services and classes of services with respect to the above dimensions and their support for aforementioned features.

4.5. New Web Paradigms Coming to Enterprises. The new functionalities and paradigms described above slowly pave their way to enterprises. On one hand, many of functionalities proposed by contemporary Web applications fit very well into quest for robust management of company knowledge. For example, wiki philosophy may be very useful in documentation creation and maintenance and may act as a supportive tool in project management; enterprise blogs can be an useful method of communication with both internal and external stakeholders (employees, shareholders, partners, customers, suppliers, potential customers). Finally, tagging (with unrestricted or partially restricted vocabularies) may be a more flexible alternative to other approaches of enterprise documents organization (such as classification and full-text indexing). At the same time, adoption of these "Enterprise 2.0" solutions is shaped by a structural conflict between openness and flexibility, typical for many modern Web-based systems, and control or rigid procedures, being landmark of contemporary enterprises.

On the other hand, we observe adoption of the paradigms related to aggregation of logic and information from multiple sources in the business scenarios. Over time more and more companies monitor and integrate information about company reputation, competitors actions and market changes from Web locations. While majority of businesses gather this information mostly for PR, marketing and strategic or tactical-level planning activities, the number of businesses using numerous integrated data sources in operational activities and the strategy of "competing on analytics" [10] is continuously growing. As more and more forms of inter-company collaboration is mediated by IT solutions, the flexible composition of logic from multiple providers (taking form of mashups, enterprise mashups, individual pipes or less loosely-coupled IT solutions) is progressing. This tendency starts to be supported by growing openness and service-orientation of major enterprise solution players (including SAP, Oracle and Microsoft). Finally, the ideas of simple, adaptive workflows combining automated activities with user involvement are becoming mainstream of research and are supposed to find their way to enterprises in closest future.

Despite this developments and buzz generated by Enterprise 2.0 solutions, majority of medium and large companies still operate multiple unintegrated or poorly integrated solutions (even if coming from the same provider) even internally. Moreover, many legacy IT software remain not well suited for or very restrictive about integration with external logic components (see for example [1]).

5. Usage. The area that recently changed the most from the point of view of people is the usage layer of the Web. It is concerned with what features of on-line applications are actually used and how. As it is an area of complex interaction between multiple systems and large number of users with very various background and



		on-line databases	search engines	on-line stores / malls	blogs	fille sharing services	news portals	forums	social bookmarking sites	project management / issue tracking	wikis	social networking sites	dating services	on-line translation services	on-line office software (e.g. Google Docs)	multimedia sharing (e.g. YouTube, Flickr)	microblogs	on-line games	Web-based instant messaging	Web-based instant VoIP	podcast
information access and search	features	■	■	■	■	■	■	░	■		■	░	░			░	░				░
information management						▓			▓	▓	▓	░			▓						
information transformation														▓							
communication					░					░		░	░				░	░	▓	▓	
collaboration and problem solving								░		░	▓				▓			░			
entertainment						░	░	░				░	░			▓	░	▓	░	░	░
education and self-development			░				░	░								░					░
business and transactions				▓																	
rhythm ([s]ynchronous, [a]synchronous, [m]ixed)	medium	a	a	a	a	a	a	m	a	m	a	a	m	s	m	a	m	s	s	s	a
bandwidth consumption ([l]ow, [h]igh)		l	l	l	l	h	l	l	l	l	l	l	l	l	l	h	l	l	l	h	h
format ([t]ext, [m]ultimedia)		t	t	t	t	t	t	t	t	t	t	t	t	t	t	m	t	m	t	m	m
permanence ([p]ersistent, [e]phemeral)		p	e	p	p	p	p	p	p	p	p	p	p	e	p	p	p	e	e	e	p

Fig. 4.4. *Examples of typical services on contemporary Web*

objectives, the "macro" impact of individual applications and their features at the social and economical level may be very hard to derive from their micro properties [15].

5.1. General Directions in Web Usage Evolution. Seen from somehow bigger distance, the Web evolves into a number of general directions. Typically this evolution means that new areas, that existed previously in an embryonic form, become mainstream of on-line businesses. At the same time, many of previously mainstream usage patterns still remain popular in specific types of services and groups of users. These general directions are:

1. Growing level of user engagement in on-line activities. Activities evolve from passive (e.g. browsing information) to active (involving participation in content creation). We identified five levels of engagement in content creation: a) no participation at all, b) unconscious participation (when patterns of users behavior are used in automatic content creation, as is in case of collaborative filtering or adaptive Web sites), c) participation in simple individual activities (e.g. tagging or rating URLs, products or blog posts), d) creative individual activities (e.g. writing blog posts or comments), and e) creative social activities (such as synchronous or asynchronous creation or management of long text documents, ontologies or databases).
2. Moving from individual to social activities. Until recently, the majority of computer-based activities were "single player". Today, a lot of them can be also done in a collaborative manner. For example, we switch from individual bookmarking, playlist management, searching, and problem solving to collaborating while performing these activities.
3. Moving from one-time to continuous and incremental activities. For example book and article writing

as well as film-making were one-time activities (after being finished the result did not change). Today even books (cf. Wikibooks) are editable in a wiki way and easy to comment on. At the same time, it became cheap to publish new versions of any digital content including multimedia and research papers. Moreover, editable and reusable content allows both the same user and other people to create new, improved or mashuped-up content. It is also a general tendency to create algorithms that first approximate results and then utilize user feedback on results (acquired through both implicit and explicit feedback cycle) for continuous result improvement. This last tendency might be due to strive to solve problems that are not tractable using traditional approach.

4. Moving from asynchronous to mixture of synchronous and asynchronous interaction mode. Majority of interaction on the Web used to be mediated by some content and performed in a asynchronous way. Recently we have witnessed, the rise of almost real-time communication channels (such as RSS-based monitoring of content, support for comments by many content sources and microblogging solutions), and propagation of more informal expression forms (even in public communication).
5. Moving from simple to complex activities. In the early days of the Web, typical users focused mostly on browsing content provided by other people. With progressing "read/write Web" philosophy users became involved in more interactive activities such as commenting or tagging content. However, it is only recently that crowds of users has started to be involved in much more complex activities such as on-line multimedia designing, crowdsourced R & D[50] or collaborative ontology development (in explicit way as in [28] or in implicit way, based on other collaborative actions as in case of [22]).

5.2. Areas of Life Altered by the Web. With development of the Web the part of our life activities that can be (at least partially) performed on-line significantly widened. At the same time, with constantly growing population of Internet users and expanding range of on-line functionalities, it is hard to imagine areas of life that have not been altered by popularization of the World Wide Web.

The Web has for example significantly changed both emotional and physical aspects of relationships, partnership and intimacy. Popularization and always-on mode of instant messaging and different methods of cheap on-line voice and video communication, changed the way people keep in touch with their spouses or partners, both during work hours and free time. At the same time, these communication methods support relationships between people spending a lot of time in distant locations. In parallel, common on-line activities such as Web-based sharing of artifacts (photos, music, links to interesting articles etc.), participation in on-line games or 3D worlds, exchange of digital gifts or collaborative creation, are becoming an important part of shared experiences of many contemporary couples. Finally, Web-based dating services and social networking sites support also formation of relationship, enabling search for partners both for long-lasting relationships based on romantic love and partnership, as well as for short-term, often sex-oriented relationships.

Similar changes are happening even more intensely in the area of friendship and social life. WWW enables easier participation in multiple social groups, varying from communities of practice and domain experts discussion forums, through different forms of on-line activist, charity or political communities, to various on-line multi-player games or virtual words fans. Some of such on-line groups bring together people with very specific interests, that are shared by few of their off-line colleagues, thus inciting the strong sense of belonging. In other groups the connection of members is strengthened by off-line activities they perform together. The Web has also a very significant impact on people reputation and status, because it works like archive of large part of our social activities. This impact is limited not only to what a person did or said on-line, but also on-line gossip or word-of-mouth about him/her. Moreover, the impact of on-line reputation is not limited to on-line activities. More and more companies skim through social media services while recruiting new employees. It is to be noted that the importance of on-line status and reputation is one key drivers of a number of collaborative efforts such as knowledge-exchange forums, open-source communities or Wikipedia (with expertise-based status), and social networking sites (with number of connections being one of elements of status).

While the education systems tend to adapt slowly to progressing "internetization" of our lifes, education, self-development and socialization has been significantly alerted by on-line services. They totally changed the way one can acquire information, thus engendering need for capabilities related to filtering, understanding and merging facts from multiple sources. The philosophies of distance and life-long learning became more feasible

[50] Examples include system that support design (e.g. in footwear or t-shirts companies such as Threadless and RYZ), and systems that support management of different product and service ideas (with examples coming from Dell, Starbucks and Salesforce; see: `http://www.readwriteweb.com/archives/ideascale_launch.php`).

because of development of e-learning (both involving teachers and using solely on-line resources). A plethora of e-learning solutions gives students more interactivity, better adaptation to their learning style and different ways of team learning. Moreover, as the Web is a very information-intensive space, even its everyday usage may be considered a type of self-development experiences.

The development of the Web have huge impact not only on development and socialization of oneself, but also on parenthood, i. e. on socialization of children. One one hand it is an extensive source of information and a communication channel joining with other people with parenthood experience (which is especially important for people with specific problems with their children such as rare diseases). On the other hand, the Web with its advantages and dangers is one of topics that need to be handled by parents in socialization process. The importance of wise parental education in this area is constantly underlined by a number of social groups both supporting IT capabilities development in children, and fighting different types of on-line abuse.

Majority of already described changes are reflected in the way people work in modern organizations. Information sources, methods of contacting other employees, customers and other business entities, approaches to sharing knowledge as well as the percentage of time one works on-line (including partial or even full-time tele-work) has changed dramatically in recent years. The arrival of Web 2.0 and Enterprise 2.0 functionalities to more and more companies[51] suggests that this revolution is not over yet. On the other hand, the symmetric change in how people interact with companies providing goods and services can be observed. It concerns the way we select products (changing mostly due to more accessible and searchable information, better price comparability and wider access to user experience stories), the way transactions are performed (on-line orders, Web-based providers contact, Web-based access to digital goods and services) and the way support and maintenance is delivered (on-line manuals, Web-based support, access to other users community, downloadable updates of software and firmware).

The Web has also a direct impact on ways of spending free time. It promotes a number of on-line individual and social hobbies (such as watching videos or playing on-line games). It supports off-line hobbies by giving wider access to information and to communities of people interested it. One of free time areas that are influenced in these both ways is related to accessing cultural heritage. The Web supports both on-line culture access (by providing virtual museums, live concerts and by including famous places in virtual worlds), and gives information about possible off-line activities and specific cultural items. The Web changes even the very conservative areas related to religiousness and spiritual life. However in this case it typically influences solely information-seeking activities. Finally, it is also an extensive source of information regarding physical activities, health and fitness.

5.3. How the Web Changes Social Landscape. As mentioned before, the Web has impact on almost all areas of life. While the Web introduces some brand new trends to the social life, in majority of cases it just strengthens the tendencies observed previously.

Following the changes that happened in 19th and 20th century, the Web significantly enlarges the space of choice in all areas of life. It gives access to an enormous amount of information concerning products and services, religions, hobbies, attitudes, cultural goods and people. It implements more and more complex search and recommendation facilities for access to these information. Finally, it enables communication with people all around the world and participation in (not necessarily geographically-bounded) niche communities focused on specific topics or activities. As a result, instead of participating in one society with single, imposed culture, semantics and values, one can select to interact with a number of specific social groups with different—possibly conflicting—perceptions of the world.

Such a change also supports further increase of importance of achieved status as compared to ascribed status. Many components of ascribed status (sex, race, health condition) are invisible or mostly invisible online. On the other hand, in large on-line communities (such as open-source community [29], forums or on-line auctions) user's social status and reputation (often measured automatically, based on past interactions) are one of basic measures of trust in given user.

These tendencies have impact on growing complexity of identity of Web users. The Web enables users to have multiple roles, participate in a growing number of groups (the notion of "neighborhood" is redefined by the Web), define herself through participation in different social networks. Moreover, the Web also gives possibility of separation of identity and person. Single user may have different (not connected) and not necessarily fully truthful nor connected to real personal data identities in multiple Web sites, making user profiles a part of "impression management, self-presentation" [9]. These tendencies to purposefully construct selves, altogether

[51] See for example: `http://www.readwriteweb.com/archives/study_fast_growing_us_companie.php`.

to mostly verbal and visual format of information transmission on the Web, make Web identities exceptionally well interpretable on a ground of symbolic interactionism theory.

Way of defining and describing self on the Web was much simpler and more limited only few years ago. People shared information about themselves mostly by constructing homepages and by using signatures in e-mail, discussion groups and Usenet (with GeekCode[52] being example of concise description of some aspects of person with a restricted, concise language codes). Since then, we have observed a number of new methods of expressing self. Many home pages have evolved into Weblogs, typically offering more frequent and time-determined update about given person. Additionally, the trend to create multiple profiles in different kinds of social services (including social networks, people catalogues and search engines, gaming Web sites, dating services and corporate Web sites) is growing in importance. Many of contemporary profiles provide information encoded at least with some level of semantics (varying form lightweight and widely used microformats to fully-fledged but still rare ontology-based representations), making machine processing of the profiles a much easier task. Moreover, new profiles are now easier to create and to be connected to existing profiles thanks to technologies such as OpenID and trends related to social network and social data portability. Other tendency is related to popularization of egocentric social networks, that help define self through social position or family and friendship relations (including also relations to fake identities such as pop culture icons). New quantitative, activities-based components of on-line identities evolved in the context of Web forums (where number of posts or average score of posts became part of social status definition), specific communities (e.g. activity measures in open-source community) and electronic commerce (where quantity and percent of positive comments on previous transactions is used as a measure of trust). Finally, growing popularity of microblogging services such as Twitter introduced new, much more dynamic patterns of "continuous self-expression".

Both collective and individual identity in Internet era are much more matter of choice that some time ago. However, the Web also influences people identity indirectly by reinforcing two previously observed tendencies related to socialization: socialization by media and socialization by peer group. The Internet partially takes over the role of both traditional media and extended peer group. Its role as a medium is concerned with dynamic flow (including on-line word-of-mouth [30, 21] and viral marketing [4]) of ideas, symbols, themes and fads (collaboratively referred to as memes) within social communities and crowds. Thus, the Web acts as a catalyst of memetic processes and bottom-up popular culture development, influencing attitudes of Internet users. As observed in [30], "compared to traditional WOM, online WOM is more influential due to its speed, convenience, one-to-many reach, and its absence of face-to-face human pressure". However, this means also more quickly changing ideosphere, reflected in more dynamic and unstable user identities.

This phenomenon concerns also politics, leading to what Yochai Benkler calls networked public sphere [3] and what other have covered by buzzword of "Citizen 2.0"[53]. On one hand such on-line public sphere means that voters use much more different information sources to formulate their opinions and judge individual candidates. On the other hands, people with similar political sympathies tend to group together and to actively take part in electoral campaign. The power of on-line political communities were demonstrated by recent US elections as the victory of Barack Obama was attributed (among others reasons) to his greater on-line activity and presence in social media[54]. While on-line political activities enable better information access and public discussion, they also increase risk of manipulation thanks to personalization of message (based on both: greater possibilities of targeting, and increased number of possible communication channels and formats) or even by hard-to-cease circulation of false, defamatory statements (few examples are given in [19]). Moreover, when people are too involved into communities sharing exactly the same opinions, the real pluralism of thought is replaced by so-called "plural monocultures", inhibiting public discussion.

Contemporary World Wide Web gives users also countless possibilities of expressing themselves in more creative ways, by democratizing social institutions related to culture creation, as well as to values and attitudes promotion. With low cost and high accessibility of media production (including both textual content and simple multimedia), the Web became an oasis of amateurism with amateur actors, performers, writers, directors and editors. While it means more freedom of creation, it also makes it harder to sieve through tons of unverified content. In general, it also means that free, amateur and dynamic content replaces at least some part of paid, professional, static and verified content.

[52] `http://www.geekcode.com/geek.html`.

[53] See `http://www.slideshare.net/jessesaves/citizen-20/` for an overview.

[54] See for example: `http://www.readwriteweb.com/archives/social_media_obama_mccain_comparison.php`.

Over centuries we observed the growth of accessibility of information on other people life. It is partially because of cultural changes that make many aspects of life come out of taboo, but it is also technological progress (especially in communication technologies) that made it simpler to acquire information on others directly from them (compare letters brought by horses, air mail and phone). With popularization of social on-line services this tendency was brought to a new level. Instant messaging, e-mail, social networks (with update tracking), blogs and microblogs allow us to track multiple aspects of life of other people in real time without even a need for direct contact. As demonstrated by research social networking tools mostly help to maintain existing off-line relationships [12]. However, they enable people to keep current about much higher number of friends. This trend combined with aforementioned "continuous self-expression" leads to a phenomenon called continuous partial attention[55], with people attention continuously split between a number of activities, resulting from willingness "to be busy, to be connected, is to be alive, to be recognized, and to matter".

The trends described above are to a large extent continuation of previous social evolutions, with significant quantitative changes related to number of participants, frequency of contact, number of choices we have, or number of impression management channels. These changes are basically quantitative, but with large increase/decrease in numbers, they become in fact qualitative. For example, on-line word-of-mouth is based on graph-like structures that mimic on-line gossip. However, as on-line networks may have much more connections and content or ideas spread much quicker, the message amplification happening on-line is qualitatively different to off-line social phenomena.

Apart from extension of existing tendencies, a few new social phenomena inherent to the World Wide Web, may be observed. In the history some components of social interaction (such as social structures, networks and expectations) used to happen "behind the scenes"—they had direct impact on life of people but remained hard to observe and understand. With recent developments in the Web, some algorithms became a new element of "behind the scenes" of social interaction. It is through algorithms (often not known or known partially) that search rankings are determined and trust is measured. There are also different kinds of algorithms that suggest what products we could buy (e.g. in contextual or behavioral advertisement), what content may interest us (e.g. in collaborative filtering, personalized search results ranking or in adaptive e-learning systems) and which people would be best partners (e.g. in dating services) or friends for us (e.g. in social networking sites). Moreover, some algorithms generate new content based on some kind of statistical or logical reasoning. For examples, automated summarization and aggregation of user opinions is used as a generalized perception of specific products or businesses[56], analysis of news and value of neighborhood houses may be used in valuation of real estate[57], and natural language processing and information retrieval technologies are used by people search sites to construct people profiles from multiple dispersed facts[58]. In all of these cases, the algorithms have direct impact on perception of products, businesses, real estates and people by on-line users. Finally, many types of algorithms performing business activities (e.g. trading algorithms used in stock exchanges or "sniper" software used in on-line auctions) shape contemporary economic environment.

5.4. Paradoxes of contemporary WWW. Some changes happening on the Web have a rather paradoxical character. On one hand we observe the socialization of previously unsocial phenomena—we observe for example the democratization of creation and growing social control over media. On the other hand many activities that are clearly social off-line may be performed in partially "dessocialized" way on-line. For example many of on-line communication channels enable anonymous discussions, and some dating services implicitly support short-term, no-involvement acquaintances (often with people providing at least some fake personal information).

On one hand the Web is the space of almost unlimited choices that enables in a much more flexible way to "be oneself" both in terms of individual self and collective identities of niche communities (the Web enables preservation of folklore or specific languages, supports contact with once culture even in foreign countries, supports development of niche, "long-tail" products, media, services and communities). On the other hand it promotes uniformization at the unprecedented level, if you are not determined enough to build up your identity. WWW is strongly dominated by only a few languages and supports quick propagation of cultural patterns and

[55] See: `http://continuouspartialattention.jot.com/WikiHome`.

[56] For example Pluribo (`http://www.pluribo.com/`) automatically summarizes Amazon product reviews.

[57] For example in case of Zillow, `http://www.zillow.com/`.

[58] Examples include Pipl (`http://pipl.com/`), Spock (`http://spock.com`) and PeekYou (`http://www.peekyou.com/`).

ideas, often driven and supported by commercial activities. That is why Internet popularization programs such as *One laptop per child*[59] are accused of cultural colonialism.

On one hand, many contemporary on-line services support user creativity through discussion, modification, combination or reorganization of existing content. On the other hand, many of such activities are limited to rather mechanical and uncreative (not to say unthoughtful) copying and pasting of information, crossing the thin line between creative combination and plagiarism or generation of noisy, not understandable content.

Finally, the Web is the space of contradictory developments regarding professionalization and amateurism. On one hand, we observe a flooding of amateur content in all areas of the Web (including such capital-intensive areas as film-making[60] and such traditionally restricted areas as legislation[61]). On the other hand, the Web is the major source of income of a growing community of professionals, and we observe a continuous professionalization of technologies handling information collection, processing and search.

6. Inter-component Dynamics. The previous section analyzed separately the changes happening in four components of contemporary Web: infrastructures, content and data, functionalities, and usage. While such abstraction allows easier understanding of some processes, the forces that span multiple components needs to be profoundly studied.

6.1. Demand-driven vs Supply-driven Developments. One of most interesting questions related to inter-component dynamics are related to causality and driving forces behind the observed large-scale changes. During our analysis we identified two opposite—yet complementary—sources of motivation for formation of complex on-line systems and usage patters, leading to demand-driven and supply-driven developments.

The former consists in a series of requirements-driven relationships. Real needs of user and business are the ultimate condition of success of new proposed approaches. Thus, they have direct impact on proposed functionalities, which define requirements of both information representation methods and basic infrastructures. This is the way that majority of on-line services were created. Search engines and Web directories (with underlying infrastructure) were created to enable easier information access, Usenet and e-mail for communication purposed, peer-to-peer solutions aim at enabling easy file sharing (disregarding copyright regulations), content and presentation separation (e.g. HTML+CSS or XML+XSLT) simplifies Web page and Web applications development, RSS aims at keeping visitors current about Web site udpates, and OpenID is supposed to simplify logging into multiple services.

The latter relation is supply-driven and acts in opposite direction. Infrastructural developments lower the barriers for new forms of content representation and types of services. At the same time, better information representation supports development of more sophisticated functionalities which in turn may create new needs and habits of users, as well as new business models. As a result, the endless possibilities of combining existing and new resources, functionalities and user actions enable new, creative, complex on-line services. Thus, a number of services is just a by-product of some need-driven developments. For example infrastructure developed for Web indexing or other large-scale Web applications, promoted gigabyte size mailboxes and a development of cloud-based applications (many of which never could afford enough IT infrastructure in no cloud solutions existed). Search infrastructures enabled also to observe what information is accessed by people, allowing for example to detect flu outbreaks[62]; at the same time, they made possible large-scale empirical Web studies without own crawlers. Usenet and e-mail were successfully used to transfer large files (through peer2mail services), many peer-to-peer solutions are now used in fully legal content distribution or in VoIP communication (well-known example of Skype), XML and RSS technologies enable a myriad of services combining content from multiple locations, and OpenID makes it much easier to collect information about single person from multiple social Web locations.

These two directions are strongly complementary and support one another. Users' demand incites creation of new infrastructures, information representation methods and service interaction models (demand-driven direction). However, once they are created, they pose an opportunity for development of new services (supply-driven direction). Moreover, new services often modify users and businesses perception and engender new needs, that start another wave of innovation.

[59] `http://laptop.org/`.

[60] First feature film fully created by fans using the Web (via Massify, `http://www.massify.com/`) is planned to premier in January 2009

[61] See for examples: `http://blog.wired.com/27bstroke6/2008/03/stanford-law-pr.html`

[62] See: `http://www.google.org/flutrends/`.

6.2. Impact of Infrastructure Development on Functionalities. As we described in Section 2.2, many specific classes of features are becoming today a domain-specific infrastructure, provided by large players, profiting from economies of scale. This progress typically lowers entry barriers and operational costs for new businesses. Thus, it has positive impact on innovation and on enriches set of functionalities accessible to the users. At the same time, it poses two groups of challenges to existing businesses. On one hand, the smaller companies operating in the areas that get "infrastructuralized" typically are unable to compete with large-scale players and need to provide different kind of value-added. As a result, the whole areas becomes cannibalized by infrastructure operators (see Table 6.1). On the other hand, lower entry barriers and operation costs, as well as changing business models of companies leaving cannibalized areas lead to more aggressive competition and dynamically changing competitive environment, thus limiting expected ROI and increasing strategic risk.

Table 6.1
Areas of research and business that may be cannibalized by new infrastructures

Services	cannibalized domains
Yahoo Pipes!	commercial mashup creation tools
OpenCalais	natural language processing software, research in information extraction from text
VoIP solutions	traditional telephony
distributed storage solutions and cloud computing	ISPs
folksonomy-based content organization	Web page directories
automated on-line translation services	professional translation services

6.3. Business Models. Business models are another element of inter-component dynamics. As no sustainable services can be provided on a long term basis without a business model (defining economic feasibility of specific enterprise), they shape the development of the Web at all mentioned levels and between them. Aforementioned infrastructurization of some part of traditional value-added of on-line companies is one of challenges of today business models. However it is not the single nor the most important one.

First area that every business model needs to address is related to revenue sources. Traditional solutions is this area include sales of goods, sales of services, acquiring commission from other businesses and sales of advertisement space. Sales of goods is currently the major source of income of on-line economy in general. However only a limited number of businesses (such as on-line stores, auction platforms or virtual malls) focus on activities related to e-commerce, and another small group of on-line services sell some items (mostly hobby-related) apart from their main operations. Today's e-commerce is shaped mostly by growing accessibility of machine-processable information about customers, competitors and suppliers, better analytical tools (including data mining, business intelligence systems as well as rule-based mechanisms for automation of transactions or other business processes), and outsourcing of non-core activities (with many shops sending goods directly from their suppliers inventories through drop-shipping, and some shops outsourcing all logistics and transaction-related activities as it is in case of Zlio.com shops). At the same time, majority of goods sold on-line become commodities accessible from multiple providers. Together with better information access this trend strengthens price competition. To circumvent this dangerous, margin-cutting tendency many businesses try to provide value-added related to after-sales services (e.g. updates, insurance, warranty, support for switching to new models), combined sales of goods and services (e.g. in telecommunication area) or personalization of products (varying from simple customization of physical product as in case of Fiat 500, through products developed in co-operation with user such as t-shirts or puzzles constructed from user photos, to products that are physically identical, but differ by accompanied services or digital goods).

The area that is supposed to prosper most in years to come is related to sales of on-line services, both computerized and performed manually. While the traditional, subscription-based information services decline and will probably be limited to a series of niche markets (e.g. access to specific databases), we observe a dynamic rise in sales of infrastructural services (e.g. storage, computing, API-based search), different types of services implementing pluggable complex logic (e.g. automatic or semi-automatic translation, accounting, massive sending of paper mail or faxes), services supporting different types of analytical activities (e.g. competitive analysis, market monitoring, search engines optimization) and access to on-line software (sold in *Software as a Service* philosophy). At the same time, a major shift in pricing models can be observed in this area, from traditional one-time or subscription fees, to fees based on actual usage (e.g. used computing power or storage, number of invocations, set of used software features).

As stated before, we observe a decline of paid information-based services. At the same time, information-intensive Web sites remain among the most popular destinations on the Web. In contemporary Web their revenue sources are mostly based on commissions and sales of advertisement. Commission-based revenues are typical for services that provide transaction-oriented information, such as comparison shopping sites or flight-reservation cybermediaries. All over information-centric sites tend to include different types of advertisement. Changes that happen in on-line ads industry concern mostly support of new type of media (ads embedded in videos, Flash animations or on-line games), popularization of contextual advertisement, better personalization of served ads (with behavioral modeling, and wider access to information about visitors), and different pricing models (with payment for ads becoming more commission-like and dependent on user attaining specific Web site goals such as transaction or registration).

Second key area of business models is concerned with operating costs. In recent years we witness two cost-cutting tendencies related to outsourcing and crowdsourcing. Outsourcing is a business trend for many years, however, recent changes in pricing models (related to pay per usage or pay as you go approaches) and much easier integration with third-party functionalities (which means lower transaction costs and switching costs) made outsourcing more profitable and manageable, in the same time significantly limiting the risk of lock-in. On the contemporary WWW, companies may outsource almost every non value-adding activity, starting with storage, computing and other instrastructural services, and including many technology and business operation tasks. At the same time many activities related to content creation, assessment and organization may be outsourced to the community of users (or crowdsourced) in the spirit of Web 2.0 services. As the users often are not paid at all, paid low wages (being rather a perk than a salary), or remunerated with low-cost, high-value internally produced goods or services (e.g. better or free account, augmented storage quota, higher content modification rights), crowdsourcing may significantly lower costs of multiple activities required by businesses. However, while crowdsourcing key business tasks (e.g. some part of R&D), the companies need to resort to specific quality assurance techniques.

Finally, the third area that has significant impact on all kind of on-line business is related to acquisition and maintenance of user base. In the traditional approach each on-line service aimed at acquiring individually as many users as possible (before competitors can surpass them) and maintaining this user base thanks to network effects and user lock-in. With recent changes related to federated identity (including OpenID) that simplify registration in multiple services, continuous development and professionalization of viral marketing campaigns, accessibility of more social-networked channels (making propagation of ideas and links even easier) and better technologies handling load peaks (e.g. cloud computing) this approach becomes even more feasible. However, experiences from early years of 21st century suggest that huge user base does not guarantee success, underling importance of revenues, costs and clear value-added. Moreover, with progressing tendencies towards data and social network portability the strength of lock-in of both users and business customers is continuously decreasing. Additionally, past experiences indicate that switching costs and lock-in effects should be counted among top criteria for selection of IT solutions. Finally, with such solutions as Facebook Platform, it is also much easier to access huge user bases of existing services. All this tendencies support more organic and value-added-centric growth of audience of on-line services.

6.4. Privacy in the Big Brother's Era. Another area that contains all components of contemporary WWW is related to user privacy. Almost every activity that is performed by people on-line leaves a number of electronic traces. Each server that is involved in complex functionalities (including proxy server and enterprise proxy servers), Web analytics software, search engines and many other services collect data regarding user behaviors. In some cases these data are directly connected to user profiles, in other cases they are anonymous but span multiple sessions and contain a lot of information about specific user (sometimes this information e.g. queries posted to a search engine is satisfactory to identify specific users). Moreover, in case user uses the same profiles (e.g. OpenID) in multiple locations, it is easy to connect behavior data from multiple sites. The integration is also simplified by concentration of many services in hands of a few big players (such as Yahoo and Google) that adopt integrated approach to tracking users. As a result, for example Google may merge browsing sessions of its search engines, all Web sites using Google Analytics, e-mail browsing by GMail, social activities in Blogger and in a plethora of other services owned by Google. Moreover, the rapid progress in Web usage mining and its applications gives the data owner growing insight into how to understand and take advantage of user behavior.

Collection of Web usage data is just one face of personal information on the Web. A lot of activities leave permanent and public results such (micro)blog posts, Usenet or discussion groups messages, comments in multiple forums, or created tags. Many informations are also shared in social networking services (they include not only information on given person but also on people (s)he is connected to), and other location such as user profiles (including home pages, institution pages, university students lists, and information legally required to be public). With growing machine-processability of Web content these information are much more easy to integrate, giving more complete view of specific user's hobbies, opinions, political and organizational involvement and colleagues.

6.5. Towards Ecosystem-Based Computing Paradigms. Many of tendencies described in previous sections of this paper involve complex interaction between multiple objects—complex flows of information between numerous services and people, composite software using multiple independent and dispersed logic components as well as numerous and heterogeneous information sources, complex interaction between multiple businesses resulting from growing outsourcing, and finally interaction between software components and people who may be involved in information flows and provide feedback on algorithms results.

All this tendencies converge, to form complex ecosystems involving software components (algorithms), people (individual acting on their own behalf, individual acting on behalf of organizations, intelligent crowds), different types of content and data, and different types of organizations (represented by business processes, procedures and rules; in case of governments it includes also legislation). This combination can be considered a new Web-based computing paradigm, concerning solving complex problems at the level of social processes.

There are a few characteristics specific for this computing paradigm. First of all, in this paradigm computing is a mixture of machine and human computing. The actual data and control flow is performed by a number of algorithmic "black boxes". Majority of them are automated, but some may be contain manually performed logic, combine manual and automated activities. It is to be noted, that these black boxes, that are composed to obtain complex workflows, may also consist of multiple embedded logic components.

Secondly, in this paradigm computing is an area of constant changes. They concern both changing internal (hidden) logic of components and changing composition of components. For example, while the basic functionality of search engines does not change and the syntax changes rarely (with backwards compatibility), used search algorithms continuously evolve. At the same time, mashups are continuously change, infomediaries and meta-search engines include more information sources, and user-created pipes can be modified within moments (when needed). Moreover, the solutions that change composition of on-line logic according to user or business process needs and past performance of specific services are around the corner. It is to be also noted that, as results of multiple logic components and workflows are stored and publicly available, many complex flow are implicit and not designed by anyone. For example whenever content resulting from some text mining or data extraction activity is stored, it is next indexed by general purposed search engines and can be included in some search-based scientific or market research workflows. All these characteristics result in computing which is distributed not only at the level of computing power (which is assured for example by cloud-based solutions), but also at the level of logic (multiple competing workflows performing similar but not identical activities can be performed in parallel, combined, compared, used to create new workflows). On the other hand, it means that results of such complex flows are not deterministic.

Finally, new paradigm of computing that we observe is not limited to flow of data and control between multiple logic components. Majority of both automated and manual tasks performed on-line have their business context. For example, many activities create legal obligations and cause money flows. On the other hand, business rules—that may depend on internal company conditions—are a an important component of control flow. For example, product search activities may end up by a transaction provided that product price is exceptionally low and company has enough of stock space at the moment of planned delivery.

7. Conclusion. In this paper we presented a bird's-eye view of changes that has happened recently at the WWW infrastructure, resources, functionalities and usage areas, varying from very technical developments to social changes that follow. We started by analyzing separately each of these components of contemporary World Wide Web, and then moved on to dependencies and relations between them. At the final part of this article we shortly presented how convergence of described changes leads to new computing paradigm, combining large variable of dynamically changing logic components with human participation and business perspective.

REFERENCES

[1] A. All and T. Byrne, *Still a big gap between reality, wishes for web 2.0.* `http://www.itbusinessedge.com/item/?ci=44015`, November 2008.

[2] V. Anupam, J. Freire, B. Kumar, and D. Lieuwen, *Automating web navigation with the webvcr*, in 9th International Conference on World Wide Web, 2000, pp. 503–517.

[3] Y. Benkler, *The Wealth of Networks: How Social Production Transforms Markets and Freedom*, Yale University Press, 2006.

[4] P. Blackshaw and M. Nazzaro, *Consumer-generated media (cgm) 101: Word-of-mouth in the age of the web-fortified consumer*, tech. report, Nielsen BuzzMetrics, 2006.

[5] U. Bojars, J. G. Breslin, V. Peristeras, G. Tummarello, and S. Decker, *Interlinking the social web with semantics*, IEEE Intelligent Systems, 23 (2008), pp. 29–40.

[6] J. Borges and M. Levene, *Data mining of user navigation patterns*, in Workshop on Web Usage Analysis and User Profiling, 1999, pp. 31–36.

[7] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, *Bigtable: a distributed storage system for structured data*, in OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation, Berkeley, CA, USA, 2006, USENIX Association, pp. 205–218.

[8] M. Cooper, *Accessibility of emerging rich web technologies: web 2.0 and the semantic web*, in W4A '07: Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A), New York, NY, USA, 2007, ACM, pp. 93–98.

[9] danah m. boyd and N. B. Ellison, *Social network sites: Definition, history, and scholarship*, Journal of Computer-Mediated Communication, 13 (2007), p. article 11.

[10] T. H. Davenport, *Competing on analytics*, Harvard Business Review, (2006).

[11] J. Donath, *Sociable media (prepared for the encyclopedia of human-computer interaction).* `http://smg.media.mit.edu/papers/Donath/SociableMedia.encyclopedia.pdf`, April 2004.

[12] N. Ellison, C. Steinfield, and C. Lampe, *The benefits of facebook"friends": Exploring the relationship between college students' use of online social networks and social capital*, Journal of Computer-Mediated Communication, 12 (2007), p. article 1.

[13] M. Franklin, A. Halevy, and D. Maier, *From databases to dataspaces: a new abstraction for information management*, vol. 34, New York, NY, USA, 2005, ACM, pp. 27–33.

[14] B. He and K. C.-C. Chang, *A holistic paradigm for large scale schema matching*, SIGMOD Record, 33 (2004), pp. 20–25.

[15] J. Hendler, N. Shadbolt, W. Hall, T. Berners-Lee, and D. Weitzner, *Web science: an interdisciplinary approach to understanding the web*, Communications of the ACM, 51 (2008), pp. 60–69.

[16] P. Heymann, G. Koutrika, and H. Garcia-Molina, *Can social bookmarking improve web search?*, in International Conference on Web Search and Web Data Mining, 2008, pp. 195–206.

[17] J. Howe, *Wired magazine: The rise of crowdsourcing.* `http://www.wired.com/wired/archive/14.06/crowds.html`, June 2006.

[18] M. Hu and B. Liu, *Mining and summarizing customer reviews*, in KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, 2004, ACM, pp. 168–177.

[19] A. Keen, *The Cult of the Amateur: How Today's Internet is Killing Our Culture*, Doubleday Business, June 2007.

[20] R. Kosala and H. Blockeel, *Web mining research: a survey*, ACM SIGKDD Explorations Newsletter, 2 (2000), pp. 1–15.

[21] J. Leskovec, L. A. Adamic, and B. A. Huberman, *The dynamics of viral marketing*, ACM Trans. Web, 1 (2007), p. 5.

[22] M. Z. Maala, A. Delteil, and A. Azough, *A conversion process from flickr tags to rdf descriptions*, in SAW 2007: Social Aspects of the Web, D. Flejter and M. Kowalkiewicz, eds., vol. 245 of CEUR-WS, 2007.

[23] S. K. Madria, S. S. Bhowmick, W. K. Ng, and E.-P. Lim, *Research issues in web data mining*, in 1st International Conference on Data Warehousing and Knowledge Discovery, 1999, pp. 303–312.

[24] L. Manovich, *The Language of New Media*, The MIT Press, March 2002.

[25] J. Park, T. Fukuhara, I. Ohmukai, and H. Takeda, *Web content summarization using social bookmarking service*, tech. report, 2008.

[26] L. G. Reid and A. Snow-Weaver, *Wcag 2.0: a web accessibility standard for the evolving web*, in W4A '08: Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A), New York, NY, USA, 2008, ACM, pp. 109–115.

[27] F. Rose, *The life cycle of a blog post, from servers to spiders to suits – to you.* `http://www.wired.com/special_multimedia/2008/ff_secretlife_1602`, January 2007.

[28] K. Siorpaes and M. Hepp, *myontology: The marriage of collective intelligence and ontology engineering*, in Proceedings of the Workshop Bridging the Gap between Semantic Web and Web 2.0 at the ESWC 2007, LNCS, Springer, 2007.

[29] D. Stewart, *Social status in an open-source community*, American Sociological Review, 70 (2005), pp. 823–842.

[30] T. Sun, S. Youn, G. Wu, and M. Kuntaraporn, *Online word-of-mouth (or mouse): An exploration of its antecedents and consequences*, Journal of Computer-Mediated Communication, 11 (2006), p. article 11.

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 243–258. http://www.scpe.org
ISSN 1895-1767
© 2008 SCPE

STUDYING KNOWLEDGE TRANSFER WITH WEBLOGS IN SMALL AND MEDIUM ENTERPRISES: AN EXPLORATORY CASE STUDY

ALEXANDER STOCKER[*], MARKUS STROHMAIER[†] AND KLAUS TOCHTERMANN[‡]

Abstract. Weblogs are widely known as a technology that allows publishing textual content in reverse chronological order, often expressing the subjective points of view of a single or multiple weblog authors. The simplicity and autonomy of weblogs is assumed to play a fundamental role in their popularity and their ability to transform implicit knowledge into explicit forms. In recent years, enterprises began to experiment with weblogs to facilitate inter- and intraorganizational knowledge sharing. Although weblogs have been increasingly adopted in a corporate context, sound exploratory and explanatory knowledge and theories about weblog adoption practices in corporoate contexts are missing. A rich toolset of network-analytic techniques exists to analyze the vast amount of electronic traces produced by large weblog networks. However, in small and medium enterprises, electronic traces are sparse due to the lack of a critical amount of weblogs being maintained, and weblog communications are intervowen with offline exchanges. This requires researchers to adopt and develop new analytical techniques and concepts for advancing the state of research on weblogs. Our paper is intended to expand existing research on corporate weblogs by studying weblog adoption practices for knowledge transfer purposes in Small and Medium Enterprises. In this paper, we report selected findings from a case study in which a weblog was used to facilitate knowledge transfer in an SME. The overall contributions of our paper are deep insights into a single case of a weblog adoption in a small and medium enterprise and the formulation of a set of tentative hypothesis.

Key words: weblogs, small and medium enterprises, knowledge management, knowledge transfer

1. Introduction.

Weblogs enjoy great popularity establishing a well-known source of user generated content on the Web. They benefit from the current Web 2.0 trend in internet technologies and business models [20] where the focus lies on user-generated content and lightweight service based architectures. Being a 'log of the web', the term weblog, attributed to Jorn Barger, refers to websites on which entries are commonly presented in reverse chronological order [21]. Termed as Enterprise 2.0 [16] or Corporate Web 2.0 [26, 29], companies have identified an untapped potential in weblogs contributing to their business goals.

As a socio-technical object of investigation, weblogs frame a broad area for interdisciplinary research. They became a new form of 'mainstream personal communication' [24] for millions of people publishing and exchanging knowledge, thereby connecting like minded people and establishing networks of relationships. Weblogs seem ideal for experts sharing their expertise with a large audience, but they also appear suited for 'ordinary' people who want to share stories with smaller groups [30]. Exploring the motivation of bloggers on the web, [18] found that blogging is an unusually versatile medium, used for everything from spontaneously releasing emotion to supporting collaboration and community. However, there is also evidence that bloggers value sharing of their presented thoughts without getting the intensive feedback associated with other forms of communication [18]. [7] and [8] characterized blogs as a medium having limited interactivity, compared to e.g. listserv. [8] found the modal number of comments in individual blogs to be zero, indicating the low level of interaction within the majority of weblogs.

In a corporate context, weblogs enjoy popularity in the form of organizational blogs. Often, such blogs are (1) maintained by people who post in an official or semi-official capacity at an organization, (2) endorsed explicitly or implicitly by that organization, and (3) posted by a person perceived by the audience to be clearly affiliated with the organization [11]. Employees are increasingly disseminating information about their experiences and progress at work to the public [4]. From a corporate view, utilization of weblogs has even been heralded as a paradigm shift for the way companies are interacting with their customers. They provide the ability of restoring a human face to a company's self-presentation with respect to information technology extending the customer relationship [3]. Aiming towards a categorization of corporate weblogs, [33] created a taxonomy describing fields of applications and upcoming challenges for weblogs.

In an Enterprise 2.0 movement [16], companies started to adopt wikis and weblogs, supporting knowledge transfer and aiming to facilitate and improve their employees' knowledge work. Both tools entail the potential of making the practices of knowledge work and their output more visible and graspable. According to [23], knowledge transfer is the uni-directional targeted transfer of knowledge from individual A to individual B. Knowledge

[*] Know-Center, Inffeldgasse 21a, 8010 Graz, astocker@know-center.at

[†] Knowledge Management Institute, Graz University of Technology, Inffeldgasse 21a, 8010 Graz, markus.strohmaier@tugraz.at

[‡] Know-Center, Knowledge Management Institute, Graz University of Technology, Institute for Networked Media, Joanneum Research, Inffeldgasse 21a, Elisabethstrasse 20, 8010 Graz, ktochter@know-center.at

sharing is an extension to knowledge transfer, where knowledge flows in both directions, from individual A to individual B and vice versa. Corporate weblogs may contribute to codification and personalization of organizational knowledge [10]. Examining internal weblogs in project management within Microsoft, [6] identified the necessity of further empirical studies on the topic of internal corporate weblogs.

Empirical studies of weblogs from academia exploring internal corporate weblogs remain scarce, and they tend to focus on large scale enterprises, which make up just a minority of all enterprises worldwide. After a brief literature review on literature with explicit focus on weblog networks within large-scale enterprises, we will address the need for empirical inquiries concerning the adoption of weblogs within small and medium enterprises (SMEs). In small and medium enterprises, electronic traces are sparse due to the lack of a critical amount of weblogs being maintained and the weblog communications are often interwoven with offline exchanges. This circumstance requires research to adopt and develop new analytical techniques and concepts for advancing the state of the art on weblogs in small and medium enterprises. Our presented findings are based on an exploratory case study conducted in an Austria SME settled in the ICT industry and employing 50 knowledge workers. We analyze structure and properties of this internal weblog and explicitly probe its impact on knowledge transfer. Our contributions are deep insights into a single case of a weblog adoption in a small and medium enterprise and the formulation of tentative hypotheses to be tested in further studies. Finally, we conclude with a summary and a discussion on the limitation of our research.

2. Related Work. Compared to the number of scientific publications on the topic of weblogs in total, those publications focusing on internal weblogs in corporate settings are scarce. A significant reason may be the fact that it is more challenging for researchers to investigate a weblog within a corporate context. Due to the sensitivity and confidence of the published information, such weblogs are "closed corporate systems". Because of the access to critical business information published, a close relationship of the researcher towards the enterprise is an inevitable precondition.

Four exemplary publications focus on a single case within a big multinational enterprise having a large set of weblogs [12, 9, 5, 14]. Such a weblog network already owns structures and properties similar to the Blogosphere, a collective term for the population of weblogs on the Web [25]. Solely through examining electronic traces created by weblog users, interesting findings about weblogs have been reported.

To learn more about structures and properties of internal weblogs within organizations, [12] investigated the internal Blogosphere of IBM. The weblog network was visualized as a social graph based on electronic traces, where bloggers and commentators constituted the nodes while the edges symbolized the relationships between them in terms of comments and trackbacks. The authors claimed to be the first to comprehensively characterize a social network expressed by weblogs within an enterprise. They presented new techniques to model the impact of a weblog post based on its range within an organizational hierarchy using mathematical operations but leaving an empirical inquiry open.

[9] explored the social aspects of blogging within an unstated large-scale enterprise using empirical methods of research. They analyzed both motivation of blogging individuals and their practices of using weblogs. Pivotal for their analysis was the observed phenomenon that busy bloggers published almost twice as much comments within weblogs they visited than posts in their own. The authors brought to light that weblogs are able to strengthen the weak ties between bloggers. Furthermore weblogs enabled an informal mechanism to encourage disparate and widespread departments to go for a constructive contact. Weblogs provided good means for employees to establish and maintain personal networks. Busy bloggers did not only create value for themselves, but also for the medium weblog users.

The growing network of weblogs at Microsoft was investigated by [5]. They studied where, how and why employees blogged, how personal the writing was in work related blogs and what happened when blogging became a formal work objective. While Microsoft valued external customer-oriented weblogs, a lot of skepticism existed towards internal weblogs to which no clear business purpose could be attributed. Contrariwise to external weblogs, internal ones were not formally supported by the company. Employees were free to determine whether, when and for what reason they blogged. A lot of bloggers described blogging as a way of sharing passion for their work and communicating directly with others inside and outside the company. Many described blogging as a desire to reveal the human side of a company, while others used weblogs purely for documentation and organization purposes.

[14] discussed roles and challenges of weblogs in internal communication in a large-scale ICT enterprise. They identified a two-dimensional framework based on the type of internal blogs and the related modes of

communication. The authors found that blogs are employed in internal communication to fulfill strategy implementation goals and to foster informal interactions. Furthermore, they hypothesized corporate climate and corporate culture determine the success of weblog adoption. Finding a balance between formal guidance and self-efficacy seems to be inevitable. In the view of the authors blogs offer an effective means for sharing knowledge in organizations in an informal manner.

3. Research Setting. The goal of our research was to probe an internal manager weblog evolving in an Austrian ICT SME employing 50 knowledge workers. The European Union provides a recommendation for classifying SMEs: SMEs are enterprises which employ less than 250 persons and have a maximum annual turnover of 50 million EUR or 43 million EUR balance sheet total. Due to the different basic conditions in SMEs compared to those in large scale enterprises, we also assume different properties and structures of internal corporate weblogs. Our research was motivated by the lack of qualitative studies of weblogs in the context of SMEs. Taken into account that SMEs comprise the majority of all enterprises worldwide, we accentuate the relevance of our study.

We chose case study research as our preferred research technique, because the researched phenomenon, the weblog, can not be separated from its context, i. e. supporting knowledge transfer. According to [32], 'a case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident'. According to the principle 'use multiple sources of evidence' [32] different sources of information had been taken into account allowing us to address a broader range of historical, attitudinal and behavioral issues. Any findings such a case study generates are likely to be more convincing and accurate. Following Patton's recommendations [22], we chose an information-rich case providing many opportunities for learning.

We started investigating the weblog with respect to its property to facilitate the knowledge transfer between manager and employees. A comparison between content of e-mails sent by the manager to all employees and the weblog content is included. Furthermore, we had the chance to interview the manager talking about his goals and the strategies of the organization. We even received a certain amount of control over the weblog, shutting down the weblog for a short period of time. Finally we carried out a survey obtaining another set of findings. Using multiple sources of evidence enabled us to derive more accuracy and relevant hypothesis in contrast to using just a single source of data.

Together with the desktop research conducted, we were able to make the following contributions:

- We showed why a weblog was used in this particular organization and how it affected knowledge transfer. Furthermore we addressed the question of weblog adoption in terms of popularity and how to raise it.
- We studied whether present techniques from internal weblog research are applicable to weblog research in the context of SMEs.
- Researching weblogs in business settings is still lacking scientific rigor. The overall goal of this exploratory case-study was to formulate research questions and to develop tentative hypotheses describing the adoption of weblogs in SMEs.

4. Conducting the exploratory case-study.

4.1. Exploring the artefact. We began our exploration by investigating the weblog's history of creation: During a critical project meeting, the manager was reporting to all employees hourly but only for a short period of time, thereby adopting a very personal writing style. After the meeting was finished, he expressed the desire to obtain a weblog for future coverage of relevant information.

An instance of Wordpress (`http://www.wordpress.org`) (licensed under the GNU General Public License) had been installed on the Web server of the company. Wordpress provides many features, but most of them remained unused within this case: A blogroll including other weblogs or web-sites which are regularly visited by the author was missing. The manager did neither insert hyperlinks to point to interesting internal or external resources, nor post multimedia-enriched content. Communicating confidential information, this weblog was accessible from the intranet only.

We explored the weblog content from both a qualitative perspective (i. e. what did the manager communicate to employees) and a quantitative perspective (i. e. how often did the manager inform the employees). From a quantitative perspective, we measured operational metrics such as number and frequency of posts and comments. Besides communicating via the weblog, the manager used e-mail as a supplemental channel. In the case of the investigated weblog, the reader group could be limited to the population i. e. 'all employees'.

The manager mainly used the weblog to share knowledge about tasks accomplished on behalf of the represented organization. Thereby he adopted a subjective informal writing style, typical for weblogs, as [11] mentioned in their paper. The communicated information was of both strategic nature, e.g. including knowledge about contracts, challenges, partner-acquisition or presentation of decisions from strategic meetings, and operative nature, e.g. including reports from business trips and stories about the participation at various events. While information relevant for all employees was shared via the weblog, time-critical information being of particular interest to a limited group of employees was transported via personal talks, telephone calls or e-mails. Time-critical information relevant for everybody was still communicated via internal e-mails to assure the information transported reaches all receivers in time.

Table 4.1
Quantitative analysis of the manager weblog

month	number posts	number comments	min	max	avg
			time difference between posts (in days)		
May	8		0	5	1,1
June	5	1	2	14	5,6
July	9		0	7	3,7
August	3		2	21	10,3
September	2		8	18	13,0
October	1		19	19	19,0
November	2		5	24	15,0

From studying the electronic traces we detected (1) a strong decrease of published posts over time and (2) a rise in the average time difference of posts over time. Furthermore, we observed the phenomenon of only one comment being posted during the entire duration of our study. We will seek explanations in the following sections, after extending the research scope.

4.2. Extending the research scope. The analysis of internal weblogs in large-scale corporate settings can be based upon extensive network data that is electronic traces of e.g. relations between a large set of internal corporate weblogs constituted by comments, trackbacks and blogrolls. Unfortunately, techniques that can be successfully applied in large enterprises [12], including network theory and social network analysis based on electronic traces, can not be applied in the same way in SMEs. In the context of SMEs, there is often only a single or a small set of weblogs involved, which renders typical research measures of network approaches [31] such as degree or centrality of weblog networks impractical or even meaningless. Instead, it becomes more interesting how a weblog interferes and interfaces with nodes (actors) that are offline—such as the different stakeholders in an organization communicating with the weblog author. Our situation required extending the scope of analyzing purely electronic traces as done in many studies of weblogs in large scale enterprises or in the Blogosphere to including offline traces of actors, reading or interacting without authoring a weblog themselves.

In this paper we argue that especially for small and medium enterprises—though we expect the same argument to hold for large enterprises as well—traditional means of social network analysis are insufficient, due to the exclusive focus on electronic traces. Analyzing weblogs in SMEs requires methods that include the offline context. There may not be enough electronic traces to accurately understand the structure and properties of weblogs and how they may be embedded into SMEs. Therefore, phenomena which are investigated purely on the basis of electronic traces might turn out to be obvious, biased or simply wrong. Our investigated case involved just one internal weblog.

A social graph based on electronic (online) traces only depicts the ‘internal Blogosphere’ as a very simple construct. We expected commenting practices to play an important indicator for the success of a weblog in terms of popularity. By observing only one posted comment, we first assumed a very low interest of the particular weblog within its possible audience. However, we wanted to learn more about the respective weblog and therefore extended our investigation to the offline actors, as demonstrated in figure 4.1.

4.3. Conducting an experiment. Contrary to the approach from Kolari [12] and our discussion in the prior section, we emphasized that it is very useful to experience the impact of the weblog on nodes (actors) which are offline, not authoring weblogs themselves. We asked the subsequent questions:

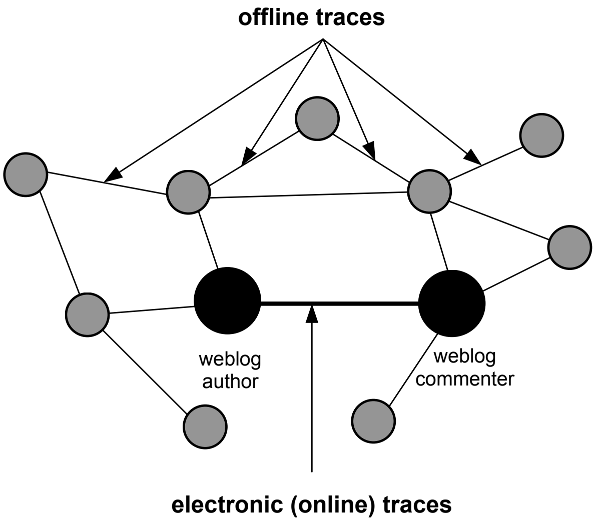


Fig. 4.1. *Social graph based on electronic (online) and non-electronic (offline) traces*

- How did different actors perceive this weblog in the context of knowledge transfer?
- What were the benefits for employees reading this weblog? Did employees ignore this weblog as a source of information, and if so, why?
- What was the rationale of just one comment being published during the time of investigation?

We setup an experiment: First we deactivated the weblog exactly seven days after a post was created. By sending an e-mail to each of the 50 employees, we asked whether they had read the recent post and were able to recall the content. Our request was repeated once to receive a higher rate of return.

14 employees in total (28%) replied to our request. 11 employees (22%) were able to basically recite the content of the past weblog post. One employee expressed that he did not read the post. Two more employees provided us with an explanation of their rationale being a nonreader. They typically read weblogs within web-based feed readers, but the respective RSS feed could not be subscribed to in this way, due to a strict firewall at place at the organization. Therefore they did not read the posts. This fact clearly depicted a goal conflict between manager and employees. Referring to [27], we assumed further goal conflicts to be a reason for weakening the intended knowledge transfer.

Analyzing the findings of our experiment, we were able to derive the following tentative hypotheses from the experiment:

- Few comments in SMEs' weblogs do not necessarily equate few readers.
- Specific IT infrastructures (firewall) are able to counteract corporate weblog practices, reducing the ability of the particular weblog to facilitate knowledge transfer.
- Studies of weblogs purely based on electronic traces may lead to biased or wrong findings. Having just a single or a small set of weblogs, it is more interesting to examine the impact of the weblog on offline nodes (actors). Social network analysis can be applied as well, but needs offline traces as input data.

4.4. Conducting a survey.

4.4.1. Survey Setup. Our first findings dealing with the actual reading behavior accentuated the need for a more detailed survey. The goal of this survey was to increase the accuracy of our findings regarding motivation of weblog readers and nonreaders. Additionally, we intended to probe to what extent the goal of the manager—using the weblog to facilitate knowledge transfer towards the employees—was achieved.

All employees who were able to remember the last weblog post during our experiment were requested via e-mail to answer six questions concerning their weblog reading practices. This population formed group A—weblog readers. All employees refusing to reply in the experiment were surveyed using a different questionnaire including further four questions. We probed their rationale of not reading the weblog, especially referring to conditions under which they would change their mind. Because we were not able to eliminate the possibility of also addressing readers, we attached the questionnaire for group A to that e-mail as well. All non readers were

Table 4.2
Weblog questionnaire(s)

Weblog Survey	
A—Readers	B—*Non* readers
A1: I read the weblog, because... *A2: How and from which location do you read the weblog?* *A3: How often do you read the weblog?* *A4: From your point of view, is commenting to the corporate weblog post reasonable?* *A5: To what extent is the manager able to improve the weblog from a technical, an organizational, and a content perspective?* *A6: Has the knowledge transfer from manager to employees been improved by the weblog compared to the previous (yes, rather yes, rather no, no)?*	*B1: I do not read the weblog because...* *B2: I would read the weblog if...* *B3: From your point of view, which particular activities are able to improve the knowledge transfer from manager to employees?* *B3.1: Do weblogs account for knowledge transfer instruments?*

finally added to group B. The qualitative data generated by the respondents' answers was then transformed into quantitative data by defining categories for the answers per question.

4.4.2. Survey Results and Interpretation. We received 40 replies (80%) of 50 possible. Altogether 20 replies were received from members of group A (readers), and another 20 from those of group B (nonreaders).

In the following, questions raised and answers given by group A will be presented. The aim of questions A1-A3 was to examine the motivation of employees reading the weblog. From an organizational perspective, further attention is paid to what extent the manager's goal of informing the employees (a) had been achieved and (b) was in fact achievable by selecting a weblog as an instrument for knowledge transfer.

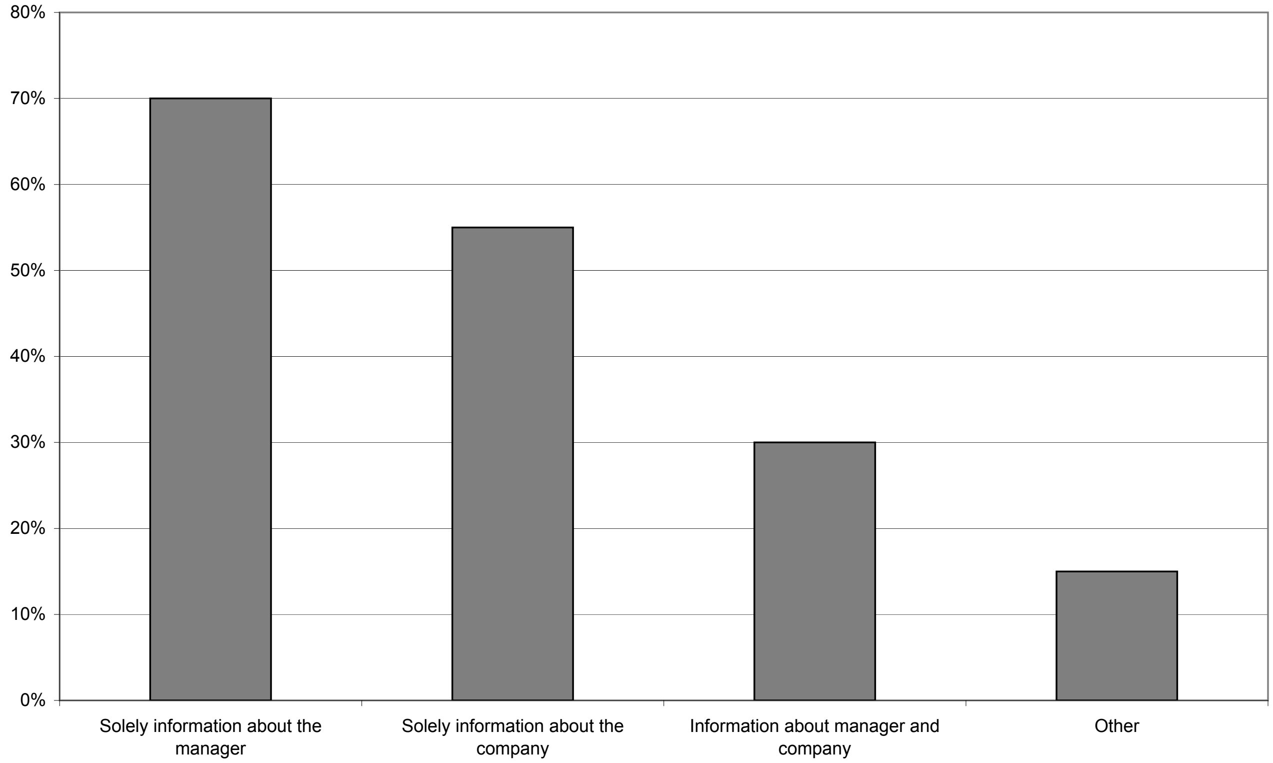


Fig. 4.2. *Motivation to read the weblog*

Question A1: I read the weblog, because...
Interpretation

Almost all replying employees clearly stated their interest in the tasks the manager was carrying out. They wanted to know, what their manager is actually doing. One third stated a general interest in what was happening within and in the periphery of their organization as well. They were curious about company strategy and organizational development. Solely the knowledge provided by the manager motivated the employees to read the weblog. Even in SMEs, where face-to-face meetings are more frequent and knowledge is diffusing faster due to lacking hierarchical structures, there is a demand for such a kind of codified knowledge from a prominent knowledge barrier. It appears that making the knowledge of a manager explicit by utilizing a weblog will stimulate a group of employees to read the weblog in corporate settings.

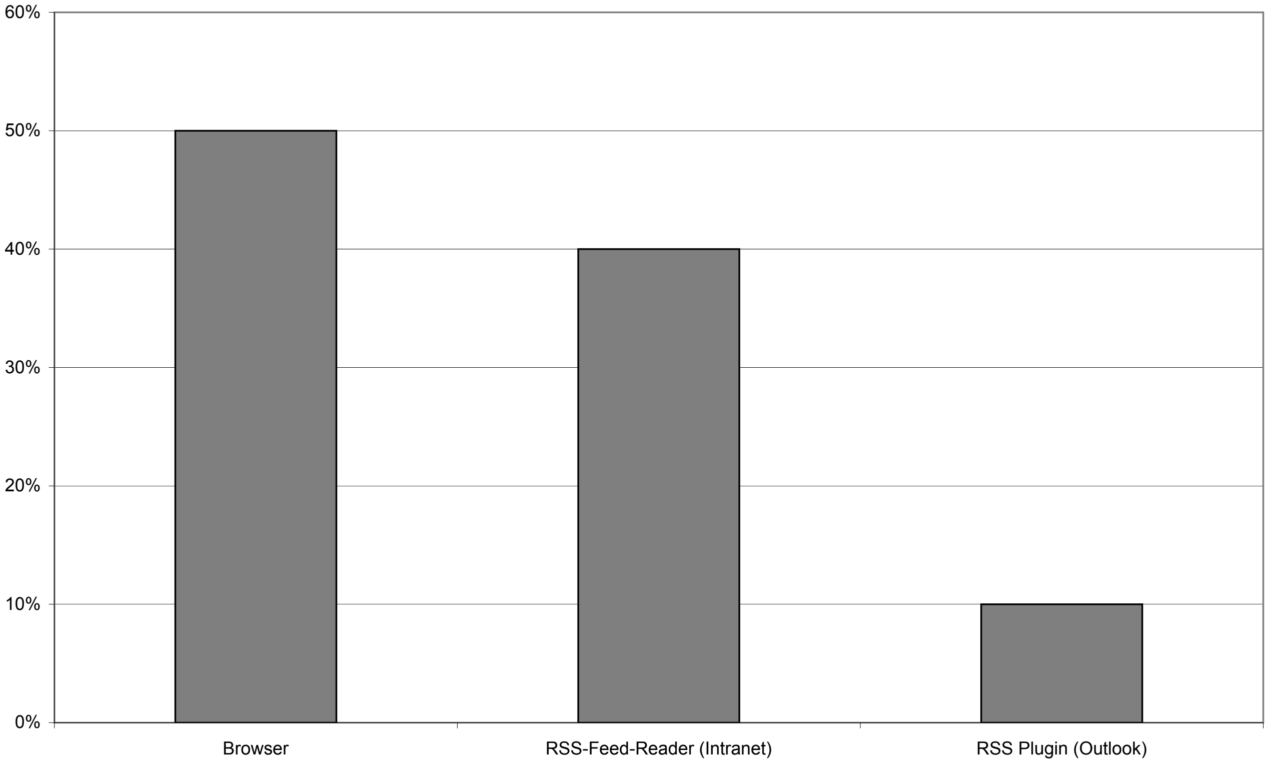


Fig. 4.3. *Tools for reading the weblog*

Question A2: How and from which location do you read the weblog?
Interpretation

Ten employees used an ordinary Web browser, explicitly mentioning Internet Explorer and Mozilla Firefox. Eight employees used RSS feed readers, while two employees went for an open source RSS plugin for Outlook. 16 employees read the weblog solely within the office and three employees explicitly addressed the access restriction, which we were also pointed out in our experiment. Because of the access restriction, employees were unable to use web-based feed readers. This constraint can conflict with the employees' weblogs reading practices. Reading weblogs by subscribing their feeds is more efficient than browsing them. However, half of the readers used a web browser to periodically scan for new posts in the explored case. We assume personal training to be crucial for establishing effective weblog practices.

Question A3: How often do you read the weblog?
Interpretation

Half of the employees browsed the weblog for newly created posts at least once a week, while five employees visited the weblog more infrequently and in broader intervals. From these findings, we assumed reading this particular weblog is more like a scan for newly created posts. Only a minor group subscribed to the RSS

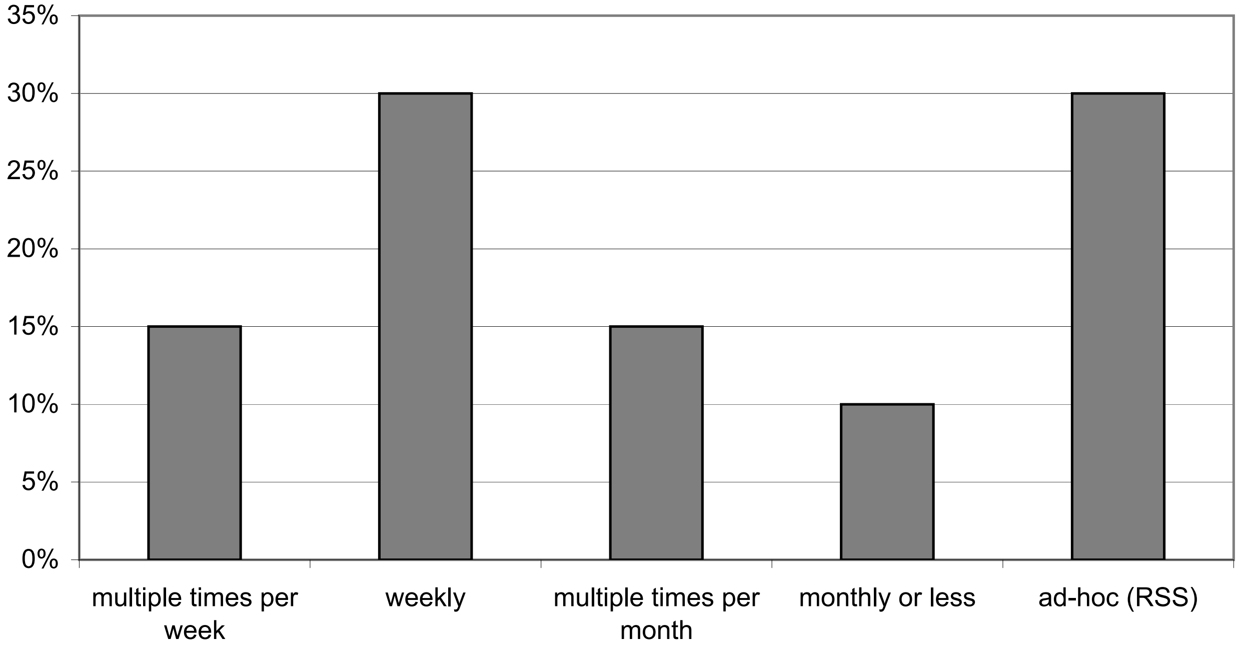


Fig. 4.4. *Weblog reading behavior*

feed, being notified after a post was published. Our results suggest that further training on (available) weblog functionality is required even in ICT companies.

The following question was aimed at exploring the reason of only one comment being posted during the time of investigation.

Question A4: From your point of view, is commenting to the corporate weblog post reasonable?

Interpretation

Eight employees positively answered this question and quoted to mention different points of view to the author, including additional information and aspects which had not been taken into consideration yet. Six employees clearly answered with 'no': The weblog was purely perceived as a unidirectional knowledge transfer medium, not a platform for sharing knowledge. The remaining employees argued that reasons both for and against comments exist. We found this question to be stated in some ambiguous way, therefore failing to deliver an answer according to our intention exploring the rationale of non-commenting within *this* particular weblog. Therefore, we try to recommend answers referring to the respective literature on virtual communities, discretionary databases and knowledge sharing.

From a virtual community research perspective and with respect to [19] the observed behavior can be termed with 'lurking', when only a marginal fraction of community members actively posts content. Lurkers constitute the majority of users in electronic forums and platforms. They for example want to remain anonymous and preserve privacy and safety, have no knowledge to offer, or simply do not feel a specific need to post.

By analyzing the social dynamics underlying knowledge sharing, [1] provide a socio-economical explanation for the identified phenomenon, the so called knowledge sharing dilemma. They treat knowledge sharing as a problem of social cooperation, manifesting in a social dilemma. In such a dilemma, individuals maximize their own pay-off for the collective's loss. The SME employees may see little reward for sharing their knowledge in the weblog and therefore they abstain.

When researching discretionary databases, analyzing the individuals' voluntary contribution to an interactive medium, [28] found discretionary information generally undersupplied. Although the technology for storing and distributing information is advancing rapidly, Thorn and Conolly see little evidence of parallel growth in the understanding of how this potential can best be harnessed. Due to their simplicity, Weblogs may reduce the

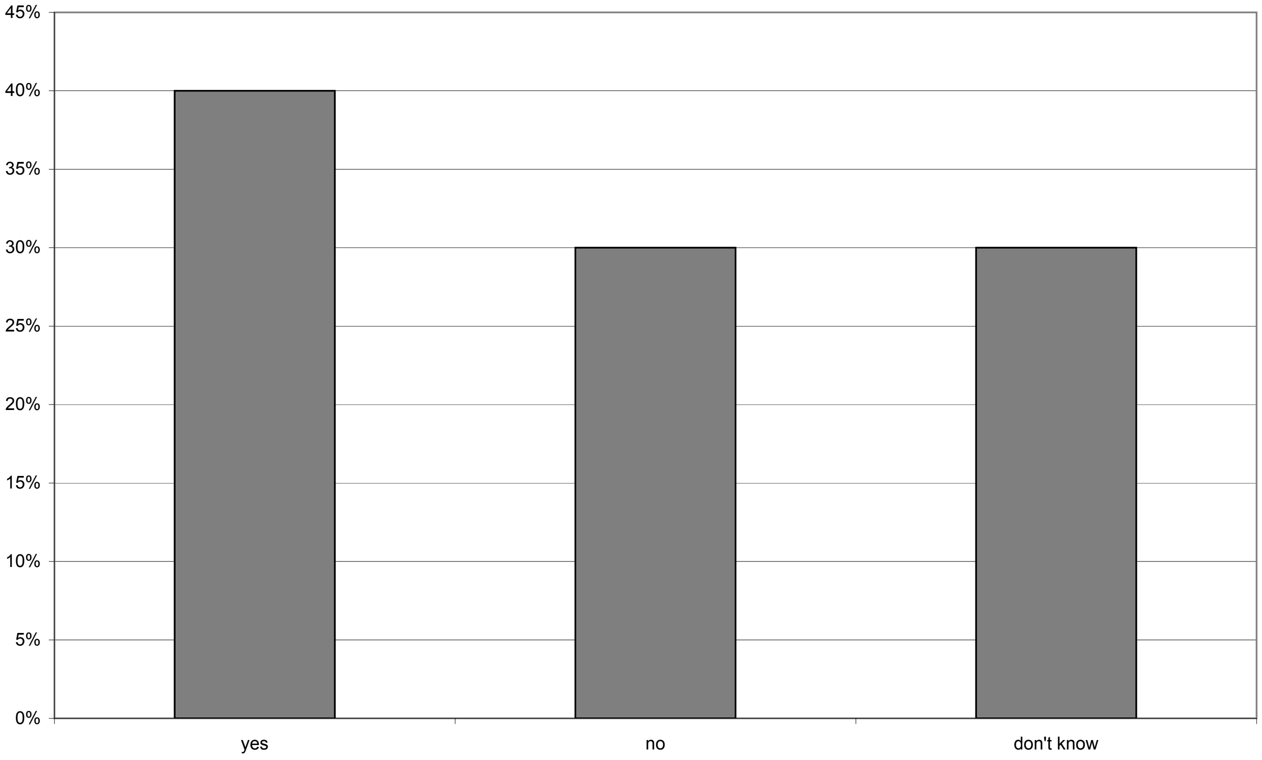


FIG. 4.5. *Commenting to a weblog post*

individual effort to contribute knowledge. However, there are still costs for individuals, related to the process of making the knowledge explicit and available in a comment, and to the social dynamic the comment may cause in the SME.

An overall question for all knowledge managers adopting weblogs deals with the rationale for users sharing their knowledge: A known motive for users is to increase their individual pay-off by sharing knowledge. The higher the value for knowledge sharing for the individual, the greater the motivation will be [1]. A lot of people conditionally cooperate in public good games. Suchlike individuals will cooperate, if others will cooperate, too. Vice versa they will defect, if others stop the cooperation. Even if participants meet again, such a behavior can be observed [13]. If people expect to retrieve useful knowledge in return, they are willing to contribute knowledge [2]. This behavior can be related to the concept of reciprocity. Knowledge sharing may even lead to a higher reputation [2] of the knowledge sharer. A high reputation can be seen as a means to advance in career, to be recognized as an expert or to retrieve a better payment. Social norms and social pressure also have an influence on the knowledge sharing practices [17]. Behavior rules enforced by sanctions of a group can arise in a scorn of the others if one will not contribute to a knowledge repository. Individuals believing their contribution valuable to others may contribute [15]. This is perceived efficacy, when somebody believes his/her individual contributions help to achieve a common goal. Furthermore, a sense of group identity and a sense of community have a positive influence on the contribution to knowledge repositories [13].

[1] suggest three possible solutions of the knowledge sharing dilemma: (1) restructuring the pay-off function, (2) increasing perceived efficacy of individual contributions, and (3) establishing group identity and promoting personal responsibility. According to the theory, the manager could have different options in our explored case: Individual pay-off can be increased by increasing the benefits perceived. For instance employees may be convinced to make comments in blogs, if they are shown that they can take part in decision-making processes by providing immediate personal feedback to the manager. Thereby they may also ease their own work and earn reputation, thus increasing their professional status. If the manager would clarify that feedback is appreciated and valuable to other employees, this may increase the perceived efficacy and lead to more frequent discussion. In principal, group identity in a SME may be higher, compared to large scale enterprises. However, communication

via a weblog may even further enhance group identity, which is beneficiary for the development of an enterprise. The manager should encourage communication via the weblog and promote a sense of belonging to the community composed of employees. Until now, no promotion activities concerning the weblog have been conducted.

Approximately half of the employees were reading the weblog. The goal of the next question was to study the barriers involved, when adopting internal weblogs in the context of SMEs.

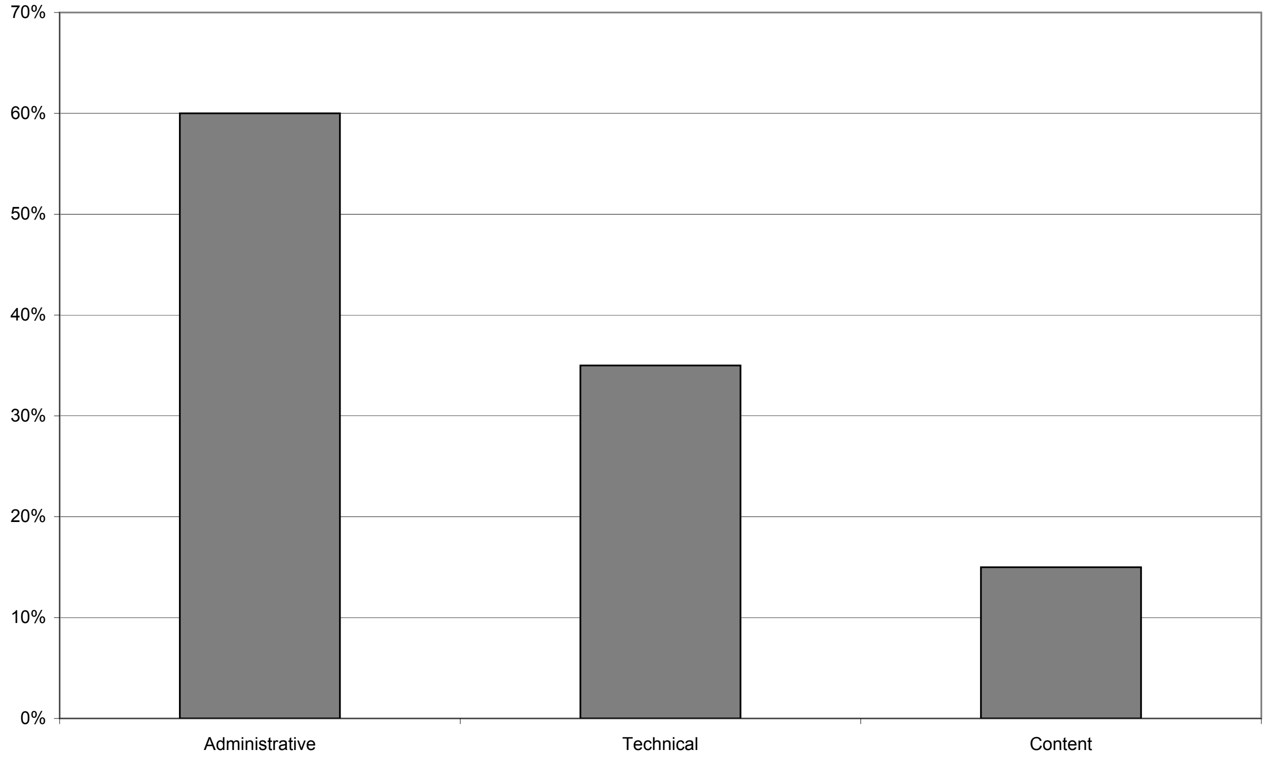


Fig. 4.6. *Improving the weblog*

Question A5: To what extent is the manager able to improve the weblog from a technical, an organizational, and a content perspective?

Interpretation

All employees reading the weblog perceived the content as appropriate for their demand of knowledge, few of them mentioned to integrate hyperlinks to (external) resources. From an administrative perspective, the most substantial criticism given by the employees dealt with the perceived low frequency of posts. Nine employees explicitly requested a higher number of posts and three employees accentuated a call for a higher frequency of comments, too. A higher number of post seems to be one necessary factor for (corporate) weblogs to be successful. By achieving a higher number of comments, because of reciprocity, more employees could be encouraged to add comments on their own, facilitating knowledge sharing. Two employees requested to utilize categories, hence clustering weblog posts and making them easier retrievable. From a technical perspective, three employees argued for making the weblog available from places outside the office. The weblog design was criticized by three employees as not being very professional.

The substantial goal of the manager was to improve knowledge transfer towards the employees. The closing question for group A addressed, whether the weblog had contributed to achieve that goal.

Question A6: Has the knowledge transfer from manager to employees been improved by the weblog compared to the previous (yes, rather yes, rather no, no)?

Interpretation

Nine employees answered 'yes', seven employees 'rather yes'. The weblog constituted a new medium for knowledge transfer from manager to employees, and the information communicated was of sufficient relevance

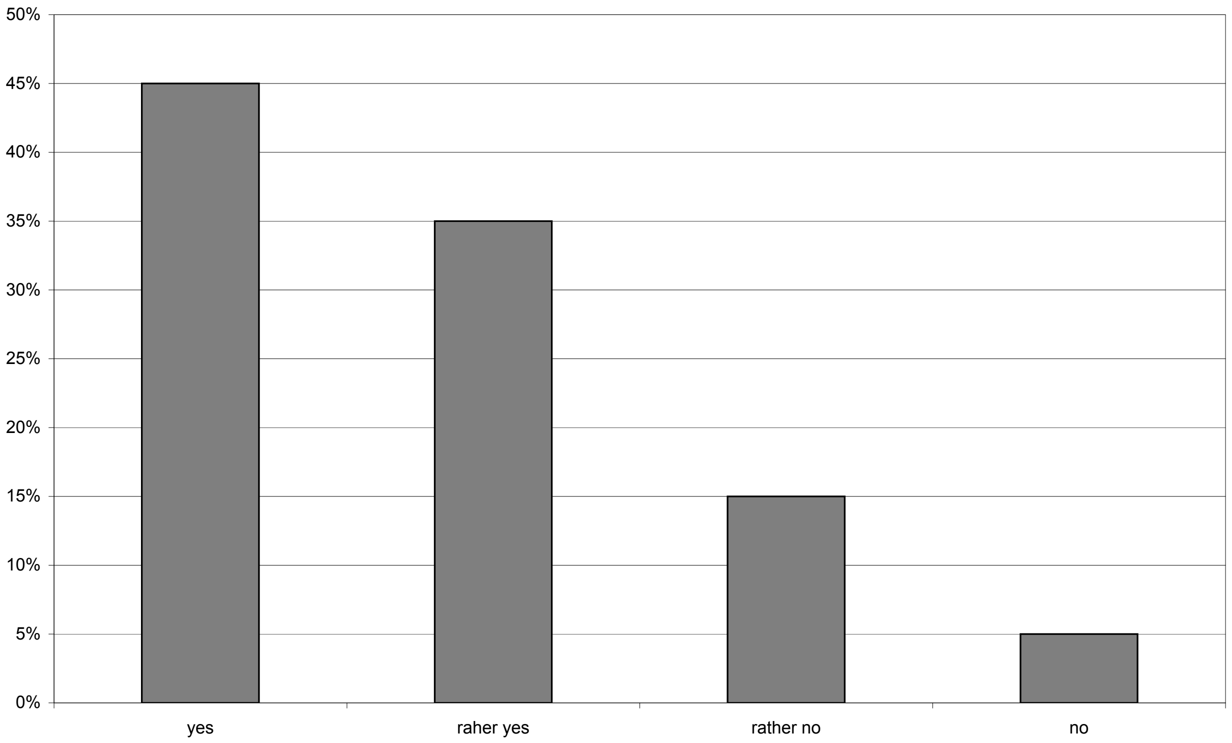


Fig. 4.7. *Better knowledge transfer through blogging*

to read the weblog. Three employees stated ‘rather no’ reasoning with the low frequency of posts, while one employee answered ‘no’. Past research on knowledge management led to a broad range of knowledge transfer instruments, which were proposed to facilitate knowledge transfer by means of organizational, sociological and technological approaches [27]. In an SME context, someone could argue that technological facilitators might be unsuidtable as opposed to organizational or sociological instruments. However, we found an internal weblog to provide a good technological facilitation of knowledge transfer, even in SMEs where the number of possible recipients is lower and hierarchies are flat, compared to larger enterprises.

Subsequent, the results of the surveyed group B are displayed. Questions B1-B2 dealt with the rationale of employees not reading the weblog.

Question B1: I do not read the weblog because...
Interpretation

The majority consisting of eight employees denied reading because they simply forgot either the existence or the URL of the weblog. Since its introduction as a new information portal, only one e-mail had been written by the author to promote the new weblog. Three employees criticized the weblogs’s lacking ability to be read via web-based feed readers. Two employees did not read weblogs at all and one employee argued a lack of time for reading activities beside the work tasks.

Weblogs provide good means to store and archive knowledge and make it easily accessible to (new) employees. Explaining the weblog’s goals to employees might help to establish it as an effective tool for knowledge transfer and / or sharing. If done so, the employees will better understand why they should read the weblog, and which individual benefit they generate by doing so. Such a status could be achieved by the help of promotion activities, which are crucial even in SMEs to sustain a weblog in its initial phase. If neglected, the weblog could remain unknown to new employees and some may even forget its existence.

Question B2: I would read the weblog if...
Interpretation

Nine employees did not see any relevance in the published content with respect to their personal work tasks, or used different channels to obtain requested information while the weblog did not provide any new insights

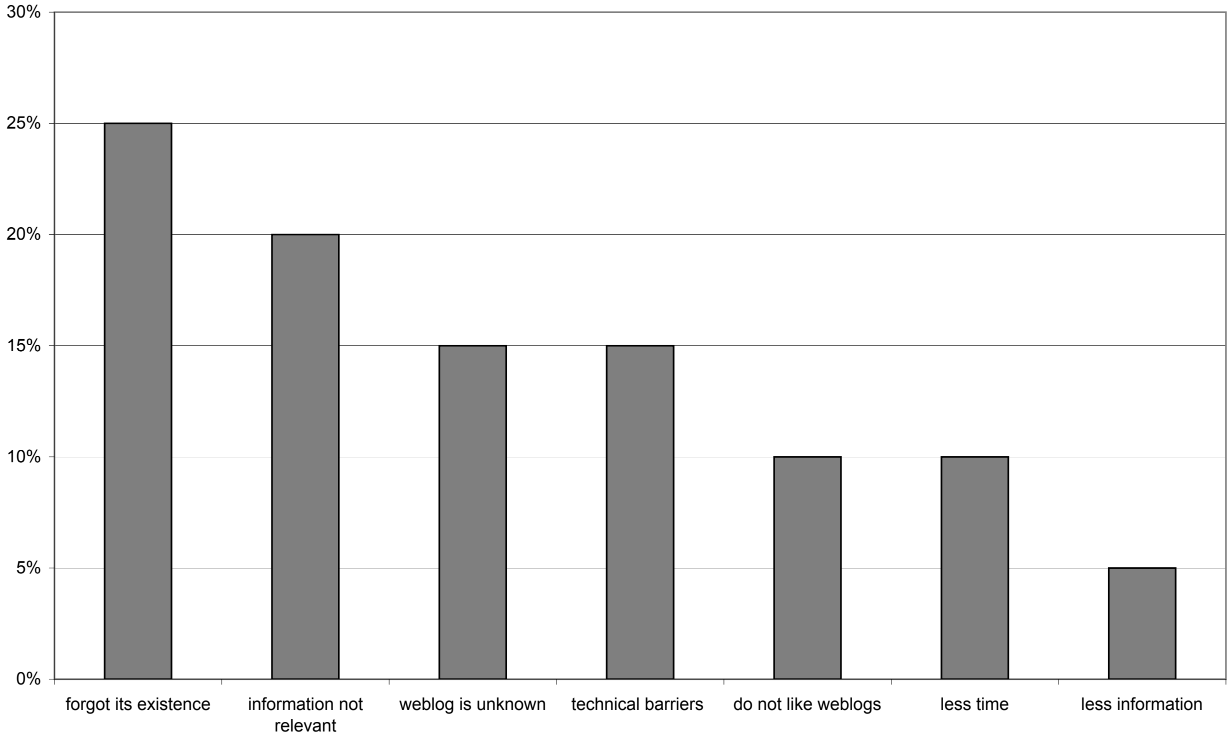


Fig. 4.8. *Arguments against reading the weblog*

to them. Six employees indicated to read the weblog if they received a notification for new posts created, for instance via email. Three employees stated to read the weblog, if it were accessible from the web allowing subscription with web-based feed readers. Due to the fact that the author of the weblog conducted almost no promotion, new employees did not learn about its existence. However, three employees were not able to provide a rationale for their non-reading behavior and promised to read the weblog in future.

One common argument for weblogs is its potential to reduce information overload and interruptions, which are often both caused by emails. However, certain employees might favor solutions based on push-mechanisms over those based on pull-mechanisms. As a result of his research in Enterprise 2.0, [16] also described knowledge workers preferring channels over portals. Adopting weblogs is different to using email and on this account affords proper training among the employees for effective usage in corporate communication.

Questions B-B3.1 addressed, whether a weblog is perceived as an instrument for knowledge transfer by the nonreaders at all. Besides that, we wanted to examine preferred knowledge transfer instruments from an employee's perspective.

Question B3: From your point of view, which particular activities are able to improve the knowledge transfer from manager to employees?

Interpretation

Prior to this survey, we assumed that *non*readers would not perceive the weblog as an instrument to facilitate knowledge transfer, but interestingly eight employees did. Besides that, personal talks, meetings, email, jour fixes and informal talks were named. Six employees placed importance on personal meetings between manager and employees. Our results show, that employees in SMEs seem to request more closeness towards their manager. On this account employees could prefer face-to-face situations, although effective and efficient tools to support internal communication, including weblogs, are available.

Question B3.1: Do weblogs account for knowledge transfer instruments?

Interpretation

More than two third of the employees acknowledged weblogs as facilitators of knowledge transfer, explicitly naming asynchrony, ease of transporting information, little effort for operation and the informal narrative

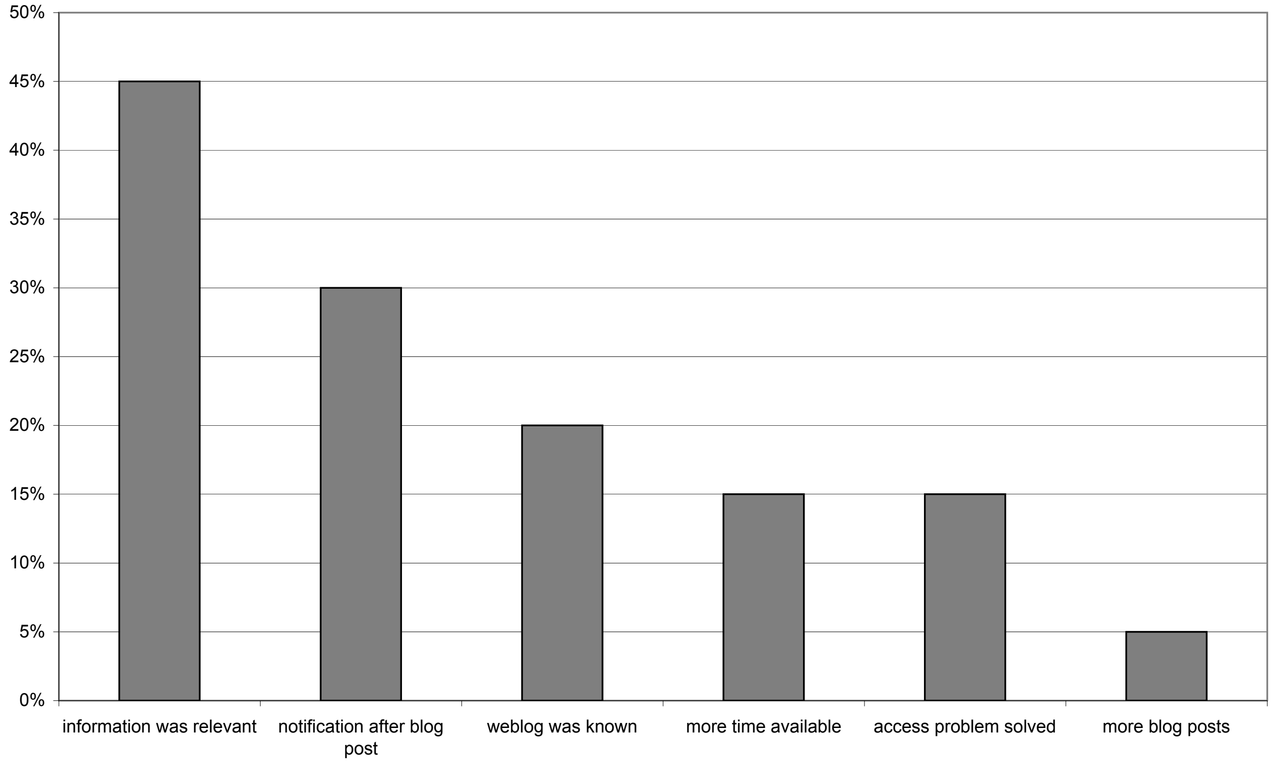


FIG. 4.9. *Motivation for reading the weblog*

style as essential criteria. Five employees negated, thereby mentioning the huge effort of retrieving relevant information. Notifications of new posts were not provided in this case, either. In addition, informal information channels seem to be available in a manageable number in SMEs. Moreover, they are easily accessible by anybody, rendering information communicated via the weblog unnecessary. Furthermore, information relevant for daily work assignments was not published. Weblogs seem effective, if people are capable to effectively use them. However, this may require intensive personal training both technical to operate the weblog and practical to accurately use the weblog.

Summarizing our findings, we derived the following tentative hypotheses for validation in further studies:

- Weblogs will be read, if they provide sufficiently interesting content that is not available from alternative sources.
- The frequency of posts illustrates a key factor for weblog success in terms of popularity. A low frequency constitutes a barrier to perceive the weblog as a knowledge transfer instrument.
- Commenting to weblog posts leads to a change of the knowledge workers' perception of the weblog as a pure information portal, hence facilitating knowledge sharing.
- Lacking skills and personal weblog practices lead to an ineffective utilization of weblogs in terms of knowledge transfer, e.g when employees demand notification features that are available but unknown to them.
- Weblogs require training, both in functions and practices on the side of the blogger, as well as on the side of the readers in corporate settings to sustain effectiveness and efficiency.
- Access restrictions regarding tools and/or location will conflict with weblog reading practices, potentially resulting in dissatisfaction.
- Weblogs have to be promoted by the authors to effectively use them as facilitators of knowledge transfer.
- Internal weblogs in SMEs are able to improve knowledge transfer in principle.
- Employees will have limited desire to read the weblog if they perceive the relevance of the published content too low with respect to their daily work assignments.

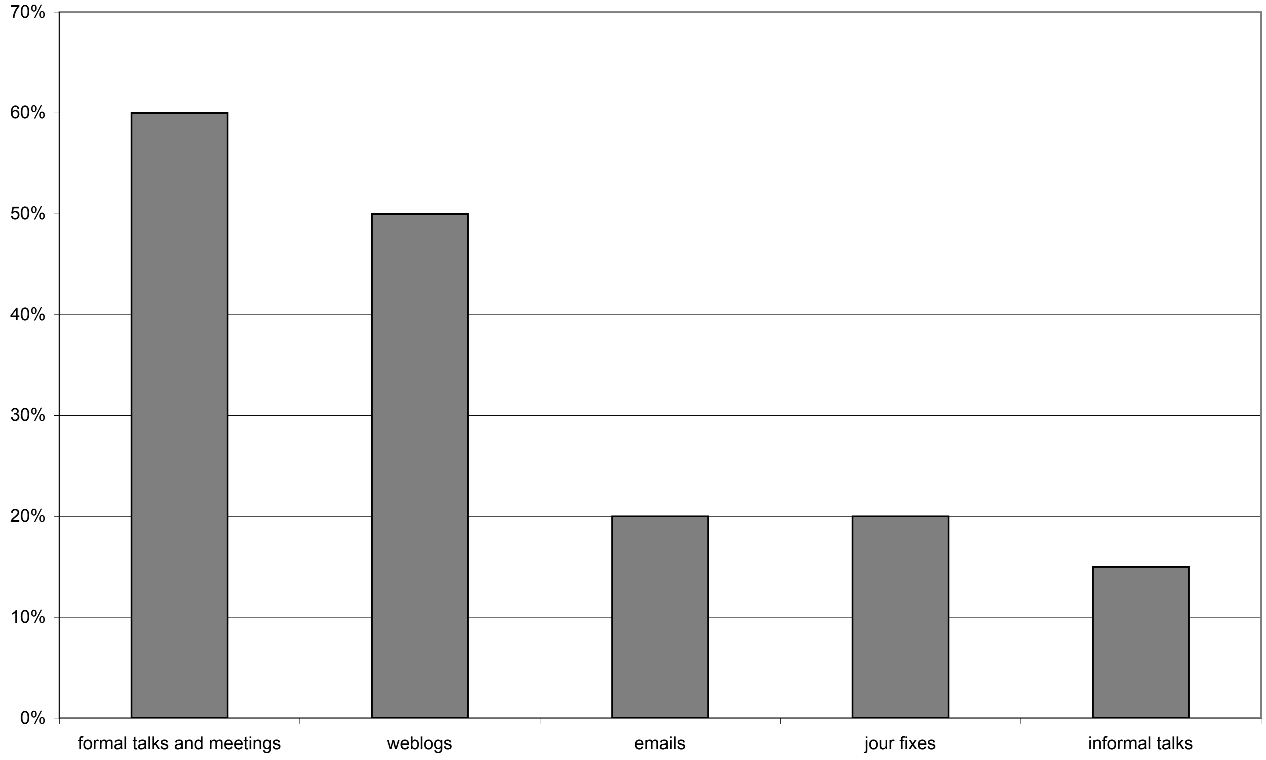


Fig. 4.10. *Instruments improving knowledge transfer*

5. Limitation of research and future work. The motivation for our single-case study was based on the fact that known preliminary academic case-studies focused on large-scale enterprises, but most of the enterprises worldwide are made up of SMEs. We intended to advance weblog research to an SME context, referring to their large population.

However, one limitation of the findings generated by our study is noteworthy: First of all, data for deriving our hypotheses was generated by only one weblog in one SME. Single-case studies provide limited utility for generalization. However, unlike surveys, case studies do not make inferences about a population (or universe) on the basis of empirical data collected about a sample [32]. In contrast to methods based on statistical generalization, case studies do not reason about the selected cases as being sampling units. Individual cases are to be selected as a laboratory investigator selects the topic of a new experiment [32]. If we had conducted a multiple-case study, the developed tentative hypotheses would have a stronger basis, allowing replication of findings. Keeping that in mind, we intend to test the hypotheses derived within further case studies to investigate whether corroboration may be achieved.

6. Conclusion. Our exploratory case study aimed at generating findings about internal weblogs in SMEs from a knowledge management perspective. The overall contributions of our paper are deep insights into a single case of a weblog adoption and the formulation of a set of tentative hypotheses. Our study constitutes a first step for more comprehensive investigations. In conclusion, we outline our contributions to organizational weblog research in a nutshell.

Unsurprisingly, it seems, that weblogs also suffer from the knowledge sharing dilemma, although through their simplicity, they will significantly reduce the cost of contributing knowledge. A high frequency of posts may constitute one key factor for weblog success in terms of popularity. However, a low number of comments does not automatically equate a low number of readers. Our results suggest, that techniques from weblog research including social network analysis, which are purely based on electronic traces, may lead to invalid findings if applied in the context of SMEs having only a single or a small set of weblogs. Our findings suggest that employees, who do not author weblogs themselves, together with their offline traces, should be explored.

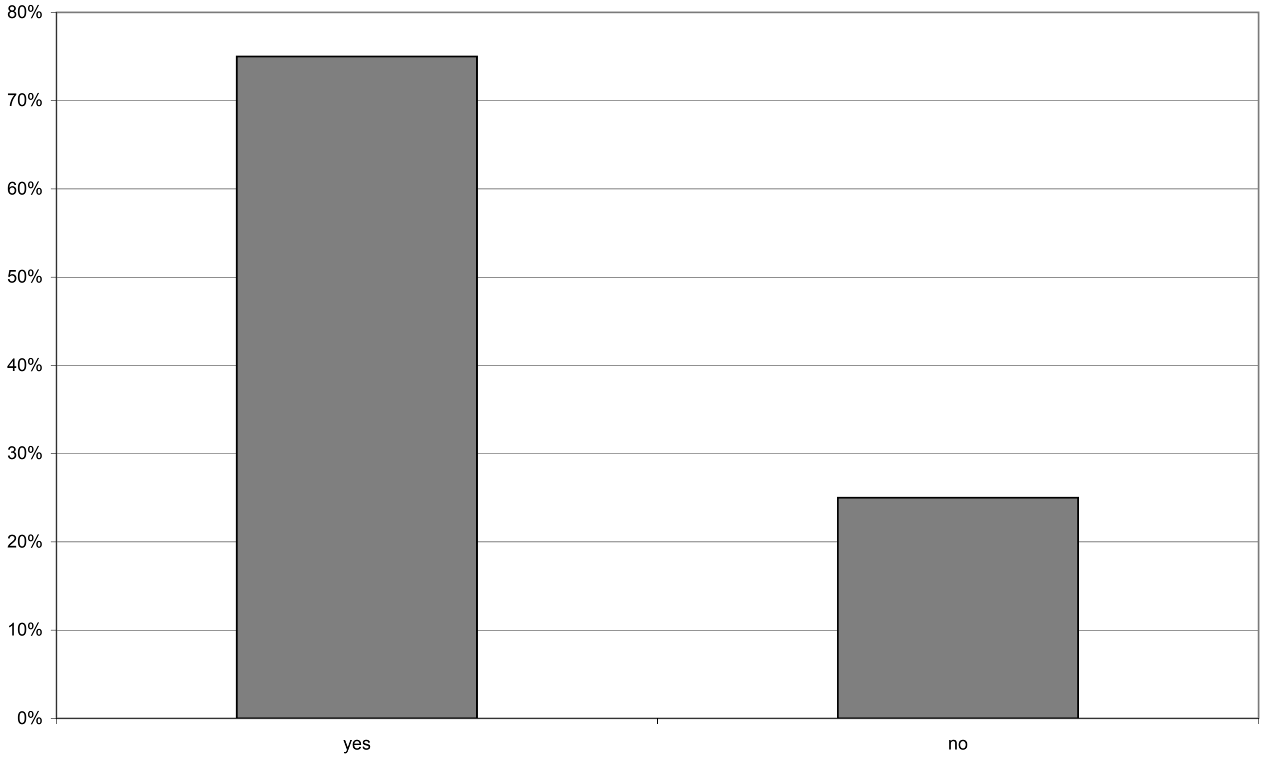


FIG. 4.11. *Are weblogs knowledge transfer instruments*

As our exploration revealed, weblogs do not run like a clockwork, but have to be actively and professionally promoted, even in SMEs where the number of employees is lower and group identity may be higher. A tighter involvement of readers posting comments might increase their perceived efficacy, thus leading to a livelier, and perhaps a more effective weblog facilitating knowledge transfer and sharing. Having more employees publishing content may also increase reciprocity, attracting more and more fellows.

We found that knowledge made explicit in a weblog by a prominent knowledge barrier (e.g. a manager) alone stimulates a high motivation for a group of people to read the weblog. It seems that employees will prefer weblogs providing information, which is of sufficient interest or direct relevance for their work assignments and not available from other channels. Explaining the goals of the weblog to employees frequently will remind them to keep in touch with the weblog.

We found that specific IT infrastructures could establish barriers, colliding with the reading practices of the employees. Our results suggest removing these obstacles through proper training on corporate weblog practices and functions, whenever possible. This will reduce possible dissatisfaction amongst the employee which is caused by ineffective usage patterns.

Acknowledgment. The Know-Center is funded within the Austrian COMET Program—Competence Centers for Excellent Technologies—under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

REFERENCES

[1] A. Cabrera and E. Cabrera, *Knowledge sharing dilemmas*, Organization Studies, 23 (2002), pp. 687–710.
[2] T. Davenport and H. Prusak, *Working Knowledge: How Organizations Manage What They Know*, Harvard Business School Press, Boston, 2000.
[3] P. Dwyer, *Building trust with corporate blogs*, in Proceedings of International Conference on Weblogs and Social Media (ICWSM-07), Boulder, Colorado, 2007.

[4] L. Efimova, *Discovering the iceberg of knowledge work: A weblog case*, in Proceedings of Organizational Knowledge, Learning and Capabilities (OKLC-04), Innsbruck, 2004.

[5] L. Efimova and J. Grudin, *Crossing boundaries: A case study of employee blogging*, in Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40), 2007.

[6] J. Grudin, *Enterprise knowledge management and emerging technologies*, in Proceedings of the 39th Hawaii International Conference on System Science (HICSS-39), 2006.

[7] M. Gumbrecht, *Blogs as 'protected space'*, in Workshop on the Weblogging Ecosystem: Aggregation, Analysis, and Dynamics: WWW-Conference, New York, 2004.

[8] S. C. Herring, L. A. Scheidt, S. Bonus, and E. Wright, *Bridging the gap: A genre analysis of weblogs*, in Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37), 2002.

[9] A. Jackson, J. Yates, and W. Orlikowski, *Corporate blogging: Building community through persistent digital talk*, in Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40), 2007.

[10] S. Kaiser and G. Müller-Seitz, *Knowledge management via a novel information technology—the case of corporate weblogs*, in Proceedings of International Conference on Knowledge Management (I-KNOW'05), Graz, 2005.

[11] T. Kelleher and B. Miller, *Organizational blogs and the human voice: Relational strategies and relational outcomes*, Journal of Computer-Mediated Communication, 11 (2006).

[12] P. Kolari, T. Finin, Y. Yesha, Y. Yesha, K. Lyons, S. Perelgut, and J. Hawkins, *On the structure, properties and utility of internal corporate weblogs*, in Proceedings of International Conference on Weblogs and Social Media (ICWSM'07), Boulder, Colorado, 2007.

[13] P. Kollock, *Social dilemmas. the anatomy of cooperation*, Annual review of Sociology, 24 (1998), pp. 1–24.

[14] M. Kosonen, K. Henttonen, and H.-K. Ellonen, *Weblogs and internal communication in a corporate environment: a case from the ict industry*, International Journal of Knowledge and Learning, 3 (2007), pp. 437–449.

[15] P. A. M. V. Lange, W. B. G. Liebrand, D. M. Messick, and H. A. M. Wilke, *Social dilemmas: Theoretical issues and research findings*, Pergammon Press, 1992, ch. Social dilemmas: The state of the art, p. 59–80.

[16] A. McAfee, *Enterprise 2.0: The dawn of emergent collaboration*, MIT Sloan Management Review, 47 (2006), pp. 21–28.

[17] R. Müller, *Knowledge Sharing and Trading on Electronic Marketplaces*, PhD thesis, Berlin University, 2005.

[18] B. A. Nardi, D. J. Schiano, M. Gumbrecht, and L. Swartz, *Why we blog*, Communication of the ACM, 47 (2004), pp. 41–46.

[19] B. Nonnecke and Preece, *Why lurkers lurk*, in Proceedings of American Conference on Information Systems (AMCIS-01), Boston, 2001.

[20] T. O'Reilly, *What is web 2.0. design patterns and business models for the next generation of software.* `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html` 2005.

[21] S. Paquet, *Personal knowledge publishing and its uses in research.* `http://www.knowledgeboard.com/item/253`

[22] M. Patton, *Qualitative Evaluation and Research Methods*, Sage Publications, Newbury Park, 2nd ed., 1990.

[23] I. Puntschart and K. Tochtermann, *Online communities and the 'un'-importance of e-moderators*, in Proceedings of Fifth International Conference on Networked Learning (NLC'06), Lancaster, 2006.

[24] A. Rosenbloom, *The blogosphere*, Communications of the ACM, 47 (2004).

[25] X. Shi, B. Tseng, and L. Adamic, *Looking at the blogosphere topology through different lenses*, in Proceedings of International Conference on Weblogs and Social Media (ICWSM'07), 2007.

[26] A. Stocker, G. Dösinger, A. U. Saaed, and C. Wagner, *The three pillars of corporate web 2.0: A model for definition*, in Proceedings of International Conference on New Media Technologies (I-MEDIA'07), Graz, 2007.

[27] M. Strohmaier, E. Yu, J. Horkoff, J. Aranda, and S. Easterbrook, *Knowledge transfer effectiveness—an agent oriented approach*, in Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS-40), 2007.

[28] B. Thorn and T. Conolly, *Discretionary data bases: A theory and some experimental findings*, Communication Research, 14 (1987), pp. 512–528.

[29] H. von Kortzfleisch, I. Mergel, S. Manouchehri, and M. Schaarschmidt, *Web 2.0*, Springer, 2008, ch. Corporate Web 2.0 applications, pp. 73–90.

[30] C. Wagner and N. Bolloju, *Supporting knowledge management in organizations with conversational technologies: Discussion forums, weblogs, and wikis*, 16 (2005), p. i–viii.

[31] S. Wassermann and K. Faust, *Social Network Analysis. Methods and Applications*, Cambridge University Press, 1994.

[32] R. Yin, *Case Study Research. Design and Methods*, Sage Publications, 1984.

[33] Zerfass, *Corporate blogs: Einsatzmöglichkeiten und herausforderungen.* `http://www.zerfass.de/Corporateblogs-AZ-270105.pdf` 2, 2005.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: February 1st, 2008
Accepted: March 21st, 2008
Extended version received: July 3rd, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 259–269. http://www.scpe.org

ISSN 1895-1767

DISCOVERING SEMANTICS IN MULTIMEDIA CONTENT USING WIKIPEDIA

ANGELA FOGAROLLI AND MARCO RONCHETTI*

Abstract. Semantic-based information retrieval is an area of ongoing work. In this paper we present a solution for giving semantic support to multimedia content information retrieval in an e-Learning environment where very often a large number of multimedia objects and information sources are used in combination. Semantic support is given through intelligent use of Wikipedia in combination with statistical Information Extraction techniques.

Key words: content retrieval and filtering: search over semi-structural Web sources, multimedia, Wikipedia, e-Learning

1. Introduction. Nowadays, organizations have to deal with information overloading. They need a way to organize and store their content and being able to easily retrieve it when necessary. Our objective is to provide a system for indexing and retrieving content based on the semantic provide by Wikipedia. Retrieving the desired content can be difficult due to the the high specifically of terms in a search task.

In our work, we are addressing the problem of accessing different kinds of unstructured or semi-structured information sources taking advantages of the semantic provided by public available resources such as Wikipedia. Furthermore using the approach we will describe in section 4 we would like to automatize the task of annotating a corpus and discover relations between annotations. Next we will use annotation in combination with textual information retrieval for determining the search context and based on it we will be able to give search suggestions and perform query expansion. Using annotation in information retrieval is not a new idea [6, 4] even in combination with ontologies [3], it has been widely used in video and image retrieval generating also a social phenomena like folksonomy [15, 13]. What is new is the use of domain independent public available semantic to automatically describe content in different kind of media.

We are applying our approach in the e-Learning context, specifically enhanced streaming video lectures (see [8, 5]) because of the peculiarity in this scenario of combining different kinds of unstructured or semi-structured sources of information. E-Learning presents many problematics in common with the business scenario in terms of content classification for its amount of information to classify and for the different contexts where a specific information can be relevant. Our target repository collects different kinds of media (video, audio, presentation slides, text documents), which can be searched and presented in combination. For each recorded event (e.g: lecture, seminar, talk, meeting) we provide not only the video but also related materials, which can consist of presentation slides, documents or Web sites the speaker points to. All the resources are temporally synchronized with the video.

An example of how a multimedia presentation of this kind looks like can be found in figure 1.1; the video with the speaker appears together with presentation slides or additional notes. Video and slides are synchronized and can be navigated by means of a temporal bar or by slide titles.

We can summarize the following five state of the art approaches to multimedia indexing and navigation:

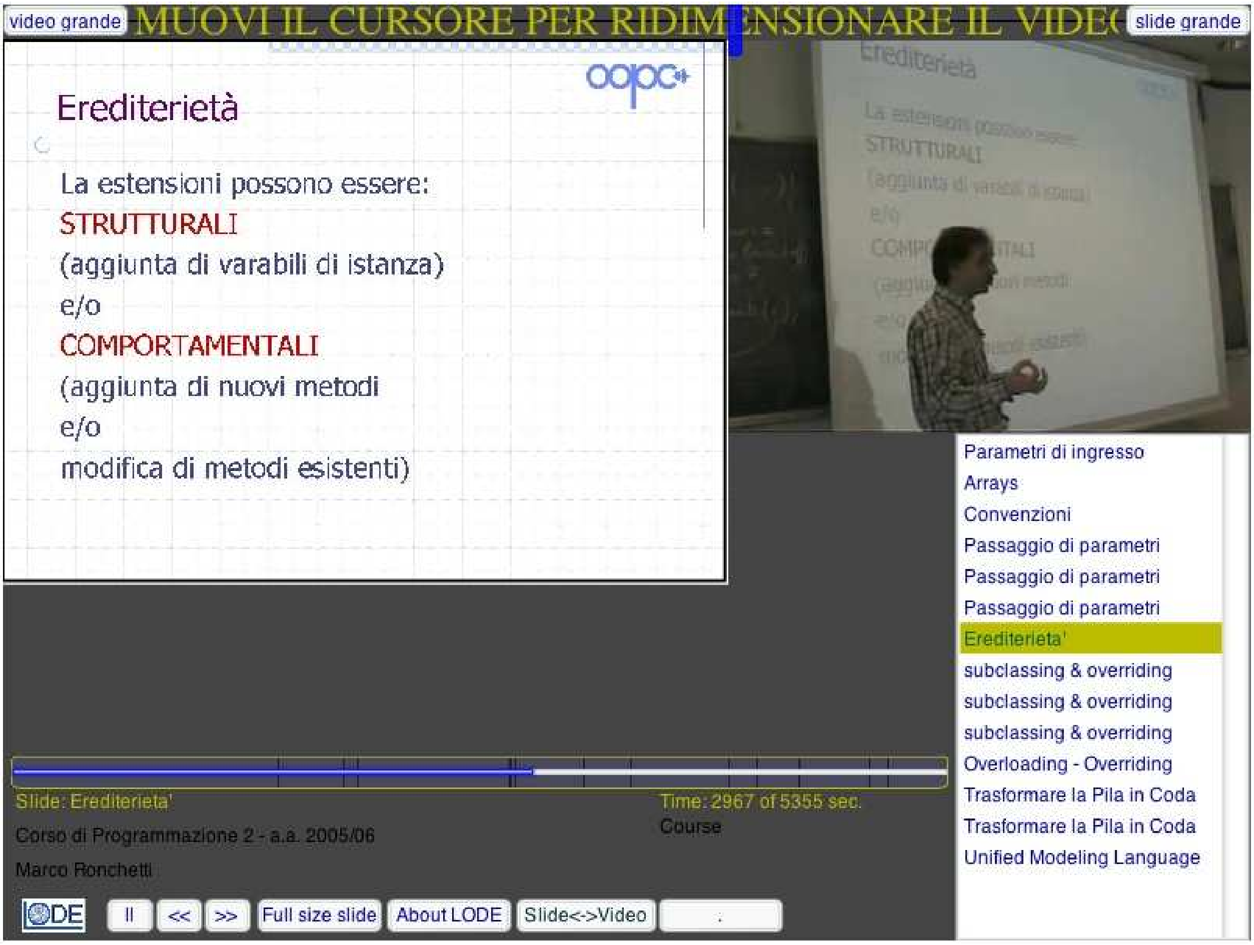
1. Use of metadata to browse keyframes.
2. Use text from speech, using transcript-based search.
3. Matching keyframes vs. querying of images. Keyframes extracted as shot representatives are used for retrieval. It requires user to locate images/other keyframes, from browsing or other search.
4. Use of semantic features. They are based upon pre-processing video or keyframes to detect features. Features can be related to ontologies.
5. Use video/image objects as queries.

We concentrate on pt. (2) and partially on pt. (3), we use the text-from-speech technique combined with a textual analysis of the speech and the event related material using Wikipedia instead of ontologies.

In Wikipedia, the concept of class and instance are not separated as in the ontological sense, due to the fact that it is not constrained to a formal model, for the reason of which it is not possible to formalize reasoning on the Wikipedia content directly.

The use of Wikipedia url as suggested in[10] for concept identification could guarantee interoperability between domain ontologies, while the extensive ongoing research effort for extracting an ontological view from

*University of Trento, Dept. of Information and Communication Tech., Via Sommarive 14, 38050 Trento, Italy {angela.fogarolli, marco.ronchetti}@unitn.it



Ereditierietà

La estensioni possono essere:
STRUTTURALI
(aggiunta di varabili di istanza)
e/o
COMPORTAMENTALI
(aggiunta di nuovi metodi
e/o
modifica di metodi esistenti)

Slide: Erediterieta'
Time: 2967 of 5355 sec.
Corso di Programmazione 2 - a.a. 2005/06
Course
Marco Ronchetti

Fig. 1.1. *An example of multimedia presentation*

Wikipedia could soon lead to the creation of an ontological view based on Wikipedia which could be the reference for many domains.

The combination of information extracted from video and related material gives a complete picture of an event, since in the real world the sum of *all* the media used by a speaker is meant to fully describe the event's topics to facilitate knowledge transfer to the audience.

In this paper we report about how we provide semantic support and unsupervised annotation of multimedia material based on information extracted from Wikipedia, rather than the usage of Semantic Web technologies (specifically without ontologies). Our approach is domain independent, and in theory it could also be applied to different use cases where there is a need for clustering or annotation of a corpus.

The structure of this paper is organized as follows: in the next section we describe the context and the motivation of our work; section 4 gives an overview of our approach. In section 5 we apply the approach described in the previous section to our use case. Finally, we discuss the directions we are planning to take regarding further work.

2. Semantics in the Web. In the Web, some collections of data containing semantic annotations (e.g. UniProt: `http://www.ebi.ac.uk/swissprot/index.html`, Ecademy: `http://www.ecademy.com`) are now available and there is a trend to semantically enable more and more Web content. Even though this trend is perceivable, there is still a huge amount of material on which these technologies have not been applied. One limiting factor for a faster adoption of Semantic Web technologies, is the difficulty to find ready-to-use conceptualizations for annotating existing material and making it Semantic Web compatible.

We explored the possibility of using Semantic Web ontologies for annotating multimedia material and for discovering and presenting to the user relations between the searched topics and other topics, based on the relationships between entities in one or more domain ontologies. We experienced difficulties in finding ontologies which cover a variety of domains, since e-Lectures can cover an unpredictable amount of domains (e.g. computer science, history, meteorology, geography, math...). In addition, the terms expressed in e-Lectures are usually *individuals* of an ontology (e.g. the term 'Collection' in a Java Programming class could be modelled as an instance of a data container class in a Java Programming ontology) and finding populated ontologies with a wide coverage of individuals to date is a big challenge, and usually requires the involvement or a knowledge engineer.

Our requirement was to find a broad, domain-independent collection of individual terms (as opposed to concepts) which are connected by relations. To the best of our knowledge, the most complete collection of this kind is Wikipedia. Wikipedia is a freely available encyclopedia which is constantly growing in size and in fame thanks to the copyleft license that allows the content to be copied, modified and redistributed as long as there is an acknowledgment of the author and the new content is published under the same license (see `http://www.wikipedia.org`). Wikipedia contains a classification of topics, organized with an hierarchy of categories and with relationships between elements. The advantage of using it is that the social collaborative network around it makes its content always up to date and it covers in details a huge amount of topics in different domains and languages. In addition it also takes into account the different possible meanings of a term through a disambiguation page.

What we can extract using Wikipedia are the relationships between topics. According to Obrst's definitions in [11], Wikipedia not only offers *weak semantic* information, such as parent-child relationships, but it also contains lexicographic relationships that—once the domain of interest is determined—can offer *medium semantic*. In Wikipedia we do not have *strong semantic*, i. e. we can not describe real-world relationships such as "a car has a minimum of four wheels" as with the usage of an ontology. We can only deduce that concepts are connected without knowing how; we can tell that one concept in one category is related to other concepts which are linked in the description of the concept itself.

In Wikipedia, the concept of class and instance are not separated as in the ontological sense, due to the fact that it is not constrained to a formal model, for the reason of which it is not possible to formalize reasoning on the Wikipedia content directly. There are projects (see section 3) that try to embed semantic inside Wikipedia extending the Wiki software used to write Wikipedia pages [16], and some others(e.g. `www.dbpedia.org` [2]) which provide an RDF representation of Wikipedia, to make its content machine-interpretable.

We use Wikipedia as a taxonomy to obtain lexicographic relationships and in combination with statistical information extraction we can deduce related concepts to the terms extracted from our corpus. In addition, since our corpus covers a representation of a part of the real world we also use the corpus itself as "training data" for domain disambiguation in Wikipedia.

There is a lot of work about extracting semantics (some is reported in section3) from Wikipedia content to build an ontological representation. We are therefore confident that even though for now we can extract information without a high semantic value in the feature, at the light and with the combination of other research effort in the area we will be able to increase the power of our approach in terms of flexibility, extension and accuracy.

3. Related Work. Wikipedia contains a vast amount of information, therefore there have been mainly two approaches for exploring its content and make it machine readable. The first approach consists in embedding semantic notations in its content [16, 7]; while the other one deals with information extraction based on the understanding of how the Wikipedia content is structured: [1, 14, 17, 12, 19].

The SemanticWikipedia project [16] is an initiative that invites Wikipedia authors to add semantic tags to their articles in order to make them machine interpretable. The wiki software behind Wikipedia(MediaWiki [7]), itself enables authors to represent structured information in an attribute-value notation, which is rendered inside a wiki page by means of an associated template.

The second main stream of Wikipedia related work is on automatically extract knowledge from the Wikipedia content as in [1, 14, 17, 12, 19].

DBpedia [1]is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia offers sophisticated queries against Wikipedia and to other linked datasets on the Web. The DBpedia dataset describes 1,950,000 "things", including at least 80,000 persons, 70,000 places, 35,000 music albums, 12,000 films. It contains 657,000 links to images, 1,600,000 links to relevant external web pages and 440,000 external links into other RDF datasets. Altogether, the DBpedia dataset consists of around 103 million RDF triples. DBpedia extracts [2] RDF triples from Wikipedia informations presented in the page templates such as infoboxes and hyperlinks.

Yago [14] is a knowledge base which extends the relationships of DBpedia extending the standard RDF notation. At December 2007, Yago contained over 1.7 million entities (like persons, organizations, cities, etc.) A YAGO-query consists of multiple lines (conditions). Each line contains one entity, a relation and another entity.

DBpedia or Yago could replaced Wikipedia as a source of knowledge in our semantic discovery approach, although at the time of this writing these knowledge bases contain only entities (such as person and places) and not abstract concepts as the one we have in e-Learning material. In addition we don't know a priori with which properties a term a can be searched, so in our domain replacing Wikipidia free-text would not be beneficial.

ISOLDE [17] is a system for deriving a domain ontologies using named-entity tagger on a corpus and combining the extracted information with Wikipedia and Wiktionary. The results shows that this kind of approach works better with semi-structure information such as dictionaries.

KYLIN [19] is another project which aim is automatically complete the information presented in the Wikipedia infoboxes analyzing disambiguated text and links in Wikipedia pages.

Ponzetto et al. [12] in their work have explored information extraction on Wikipedia for creating a taxonomy containing a large amount of subsumptions, i. e. is-a relations.

4. A Semantic Discovery Approach . In this section we explain the process of extracting semantics from multimedia content. We tested the approach here described on e-Lecture presentations. An e-lecture is a multimedia presentation usually composed of a video with focus on the speaker, presentation slides and other textual documents which can be identify by the presenter as related source of information. Different media in a presentation are used for drawing a better picture of the content to be communicated.

Hence itđ's of fundamental importance to take into account the different modalities of the media. In particular, we investigate textual modality analyzing the full content of the related material such as slides or documents and the auditory modality translating it in textual which represents the most promising aspect of the data we can process. For this reason we apply automatic speech recognition (ASR) to the video soundtracks and subsequently we are interested in the STT translation (speech to text) to provide for data that can be analyzed in combination with the other textual resources such as slides, notes and other documents.

The reason of focusing on auditory and textual modality instead of visual modality is intrinsic on the nature of presentations. Unlike other domains such as movie or news, in video presentations the images in the keyframes are more or less still, usually the speaker and part of his/her presentation is captured. The scene almost never changes, the transitions being related to a change of focus (from slide to blackboard and back) or to the change of slide. So, in e-Lectures, even though low level feature recognition such as teacher gesture and facial prosody might give information about importance of certain passages, we decided not to attack this issue because we believe it would only bring us a minor added value in comparison to the knowledge we could retrieve exploring the auditory and textual modality.

Due to these considerations, we focus our research work on making better use of speech and textual content. Furthermore, relating the extracted speech and textual content with the right domain knowledge could provide another mode to tackle the semantic gap allowing more effective classification and searches on the video content.

In figure 4.1 we give a graphical illustration of the process of extracting semantic annotation from multimedia content. As input the system receives a e-Lecture presentation and based on the Wikipedia knowledge it automatically generates some descriptives labels for the multimedia content.

In the next paragraph we will describe in details how this process of automatic semantic discovering is taking place.

The explanation of our method can be split into two parts. In the first part deals with the extraction of content from multimedia lecture materials without any regards about semantics; while in the second part we go in depth in the passages which involve discovering the semantics behind the content previously extracted from the media.

So, in order to discover the semantic present in a corpus we first have to extract and identify terms from it. Once we have the list of the words contained in each unit of the collection, we can link them through the relationships we will determine through Wikipedia. In particular the first part is about Information Extraction from the multimedia content and the second focuses on describing how through Wikipedia we can annotate the material and find semantic relationships between annotations.

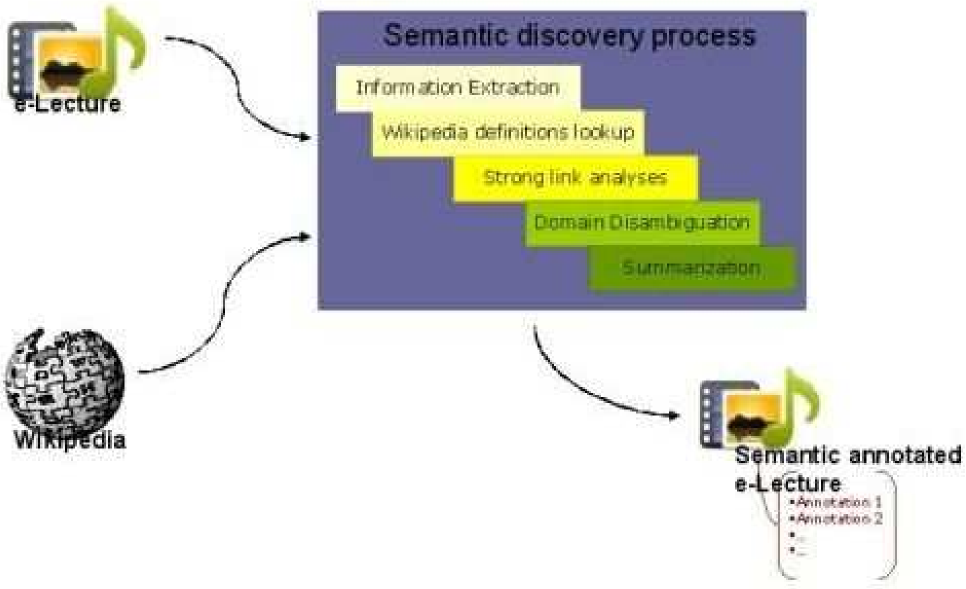


Fig. 4.1. *General Overview of the approach*

The two steps are independent in the sense that Information Extraction can be carried out in different ways while our Wikipedia module could still be used to find relations between terms. We give here an explanation of the first part only for the sake of contextualization.

4.1. Information Extraction. In this section we will extract and model the multimedia content though the analysis and combination of the textual and auditory modality. Using an out-of-the-box speech recognition tool we translate video speech into text, and we combine it with the text extracted from presentation slides and other textual information sources.

Secondly, we extracts terms for the resulting textual resources which represent the entire content of the multimedia material. We performed Information Extraction (IE) by using Lucene (`http://lucene.apache.org/`), a state of the art tool which provides Java-based indexing and search technology using a statistical approach.

Lucene had been used in the project as a search engine for querying an unstructured e- Learning repository, but since it also provides basic APIs for analyzing text, we exploited Lucene also for extracting information from our corpus. In general term extraction tools using a statistical approach basically look for repeated sequences of lexical items.

Consequently we store the extracted information in a Lucene index that we later use for information retrieval and for extracting the most important terms out of the entire e-Lecture content. We also explored a linguistic approach based on Natural Language Processing (NLP) using other state of the art tools in the area such as GATE (`http://gate.ac.uk/`) and IBM UIMA (`http://www.research.ibm.com/UIMA/` [18]), but the approach was not suited for our use case since it is language and grammar dependent. In fact, in e-Learning the material can be in different languages, and sometimes more than one language can be combined in the same event. For instance, in some cases presentation slides are written in English while the speech is delivered in another language (e.g. in Italian). As a consequence, the work needed to adapt a linguistic approach to our needs was excessive.

Moreover, story telling does not play an important role in e-Learning—at least not in the disciplines we considered—and this makes it difficult to locate and classify atomic elements in text into predefined categories for Entity Recognition. For these reasons we chose a statistical approach and we calculated a term vector for each document in our index. The term vector contains a list of terms with their frequency in a document.

In order to calculate the term vector we had to store our multimedia material into an index. Many documents can refer to the same event. For instance, we have at least two main information sources for each event: presentation slides and video transcript.

Following our multimodal view, we modeled the entire eventďs multimedia material into a single document in a index. In this way the term vector calculated through Lucene is factual.

For improving the performance of this task we are currently working on the indexing phrase and in particular in the pre-processing task (e.g. cleaning the text from Italian or English stop-words and applying different language stemmers as a filter, building categorizer for improving the quality of the raking of the extracted lecture terms.).

4.2. Semantic Extraction using Wikipedia. In this section we explain how we enhance information retrieval based on the recognition of the most important topics and the relations between them in the content of multimedia lecture material.

Understanding what a media is about before entering in its details of the search results, which would mean watching a video presentation or reading the related material, is one of the mail goal in the area of Multimedia Information Retrieval, and it could be very useful when the amount of the multimedia material grows in size. Having a way for categorizing or understanding the main concepts in the content will help in managing large multimedia repositories.

Furthermore, e-Learning users (typically students) do not have a rich understanding of the domain or of how one topic is connected to others. For this reason a tool, which has the goal of enabling access to information, should give an overview of the material content for helping end users to achieve more effective searches and acquire the needed knowledge. For example, a user looking for the term ďCollectionď in a Java programming class must find out about the different types of Collection such as ďHashMapď, ďMapď and ďSetď since these terms can also be found in the lecture material, and they all mean Collection. Understanding relationships between terms in our corpus permits also us to automatically discover the important topics of an event which can be used for unsupervised classification of the material.

Our starting point for the second phase of the semantic discovery process is the list of terms which were extracted from multimedia material (video, slides, and documents) during the Information Extraction phase described in the previous section.

In Wikipedia, we look up the most important extracted terms from our corpus. The goal of this phase is to find a Wikipedia definition page for every important term and to try to extract relations to other terms by examining the hypertextual links in the page. This is done by processing links in the page. Therefore, the term of interest is found in Wikipedia, and all the links in its page are analyzed.

Pre-requirements for describing the process of extracting semantics from the Wikipedia, are the four abstract concept described below that we will be mentioned again in the second part of the section for giving a description of the process itself.

1. *Wikipedia lookup.*
   For each extracted term we search for pages in Wikipedia which contain the term in their name. In Wikipedia every page is named by a string composed of topic name and topic domain. After that, we collect the links for every page. The search is made on a local copy of the English version of the Wikipedia database, but we could also reach the same result by downloading and parsing Wikipedia Web pages. We chose to maintain a copy of the database to increase the speed of the task.
2. *Strong link definition.*
   We define a link to be ďstrongď if the page it points to has a link back to the starting page. For instance,řRomeř and řItalyř are strongly linked since the page on Rome says that it is the capital of Italy, while the page on Italy reports that Rome is the capital of the state. A minor town located in Italy will instead have a ďweakď link with Italy, since in its page it will be declared that the town is in Italy but in the page for Italy the minor town will most likely not be mentioned. In our case, strong links are candidates for topics related to the searched term, and they will be used for giving user suggestions in query expansion and in the process of summary generation of Wikipedia definitions.
3. *Domain disambiguation.*
   A word can have multiple Wikipedia definitions because it can assume different meanings (senses) in different domains. Among the (possibly) multiple Wikipedia definitions, we choose the one which has the most link words in common with the extracted lectureďs terms. We manually checked this approach

to evaluate the accordance of the semantic expressed in the disambiguated terms with the one of the event and we find out that this is true for the majority of the cases.

4. *Annotation through Wikipedia definition summarization.*
In this last step we use the extracted strong links for every important word of an event to automatically generate a summary of the word definition in Wikipedia.The summary is generated taking all the sentences from the Wikipedia definition page in which a strong link is present; usually fifty percent of the content of original definition is selected. The summary is then used for expressing the meaning of the important term. In other words, we annotate the lecture through Wikipedia terminology, and for each term we keep a brief definition.

A graphical representation of the process is given in figure 4.2.

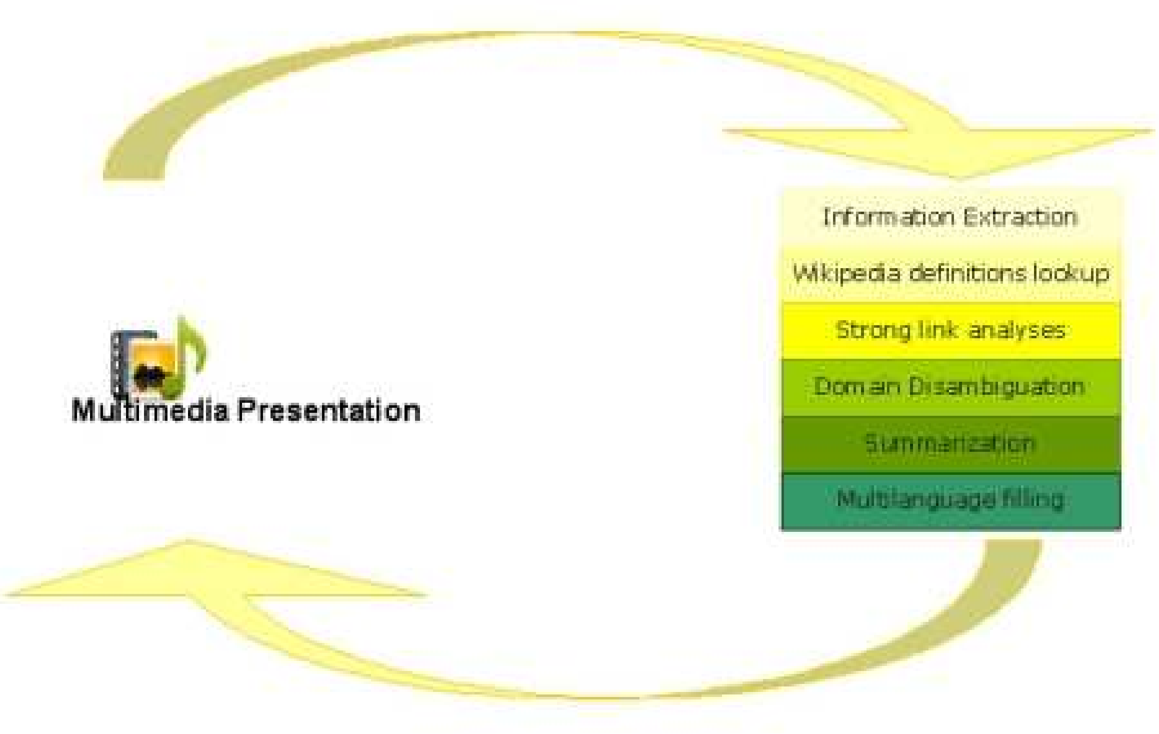


Fig. 4.2. *Semantic annotations generation process*

The figure shows that by giving e-Lectures as input and through the four steps of the semantic discovery approach we can enrich e-Lecture with semantic metadata. It follows a description of the semantic discovery process composed of four logical steps: Wikipedia definition lookup, Domain disambiguation, Strong link Analyses, and Summarization.

Suppose one of the terms extracted from the e-Lecture material is ďCollectionď, which is in the list of the extracted keywords. Consider a simplified list of extracted keywords (note the presence of words in more than one language!) follows: Elemento, Map, Tipo, Object, Method, Interface, computer science, Collection, Oggetto.

The first step of the algorithm is to search for all the pages which contain the term ďCollectionď in their name. A search in the Wikipedia database will find a relatively large number of pages that satisfy this requirement due to the different meanings the word can have. Consequently we retrieve the links of every found page. We then use the links in the disambiguation step. For instance, in the case of having a term named ¿Collectionř in the lecture material, a Wikipedia query for the word Collection returns the Wikipedia disambiguation page Collection, which points to other pages such as Collection(horse), Collection(museum), Collection(Joe Sample album), Collection(agency), Collection (computing), Collection class.

The second step consists in extracting the strong links from the candidate Wikipedia article for each definition. The strong links are used in the third step for calculating domain disambiguation, for support and for automatic summarization of the content of Wikipedia entries. So in this step we extract the strong links from all the Wikipedia pages selected during the first step. For example the strong links for Collection (computing) which is one of the pages selected during the first step, are: object-oriented, class, map, tree, set, array, list. We do the same with all the pages listed in the Wikipedia disambiguation page for collection.

The third step resolves domain ambiguity. We automatically identify the right Wikipedia definition based on the domain defined by the multimedia. So we select one page among all the ones we retrieved during step one which is in concordance with the domain of interest. The disambiguation function considers for every candidate page its strong links given by step two. In particular it looks for correspondences among the linkd's names and the keywords extracted from the corpus. The page which has the largest number of links in correspondence with the corpus terms will be considered to be the correct one and it will be used as the disambiguated term. Supposing that Collection(computing) is the Wikipedia article is the page that responds to this requirement then every time Collection will be mention in the lecture material it will be associated with the meaning of the Wikipedia article Collection(computing). The term that has been disambiguated has the same meaning in Wikipedia and in the corpus. The result of this step is the identification of the disambiguated terms with their links. In other words this step compares the strong links of every candidate Wikipedia definition with the term vector of the lecture in exam.

During the last step we create a summary for the most important words in the lecture. Each term in the lecture has a corresponding Wikipedia definition and based on the ranking in the term vector we can identify the n most important terms in the e-Lecture. The summary creation uses the extracted strong links of the most important terms. For every one of them, we download and parse the correspondent Wikipedia. From the Wikipedia page we select only the sentences which contain a strong link and the term itself. The combination of extracted sentences permits to generate a reasonably well written article summarization.

5. Applications. In this section we describe some applications where our approach can be used. Many other applications are under considerations. In particular we applied the semantic discovery approach into NEEDLE [8]- Next gEneration sEarch engine for Digital LibrariEs -. NEEDLE is an e-learning application which aims at indexing, searching and presenting structured and unstructured multimedia data. The system provides a way to search e-learning materials through a web-based search interface.

The e-Learning materials consist of video lectures and corresponding audio tracks, PowerPoint presentation slides etc. The application's main objective is to present the structured and unstructured multimedia e-Learning materials. The users could query the NEEDLE system to search for materials of their interest.

The data i. e. video lectures and slides, for NEEDLE come from the LODE [5] system. LODE is web-based application for presenting the video lectures synchronized with presentation slides. The audio content of the video lectures from the LODE system is transcribed using speech recognition and speech processing tools. The text content of the transcription and PowerPoint slides are indexed and searched using NEEDLE system.

Most of the commercial search engines only offer text based search and few also provide image search. However, there is still a need for searching video, audio, graphics etc. Commercial video hosting sites like YouTube that offer search for video actually performs the search only on the meta-data (text content describing the video) attached with the video. They do not search the video/audio content. We offer textual search on audio content using the transcript in combination with the content of the documents which come with the video such as presentation slides; moreover we enanched the search task with search suggestion based on the identification of relationship between topics in Wikipedia and we automatically extract also through Wikipedia labels for describing the most import lecture's topics.

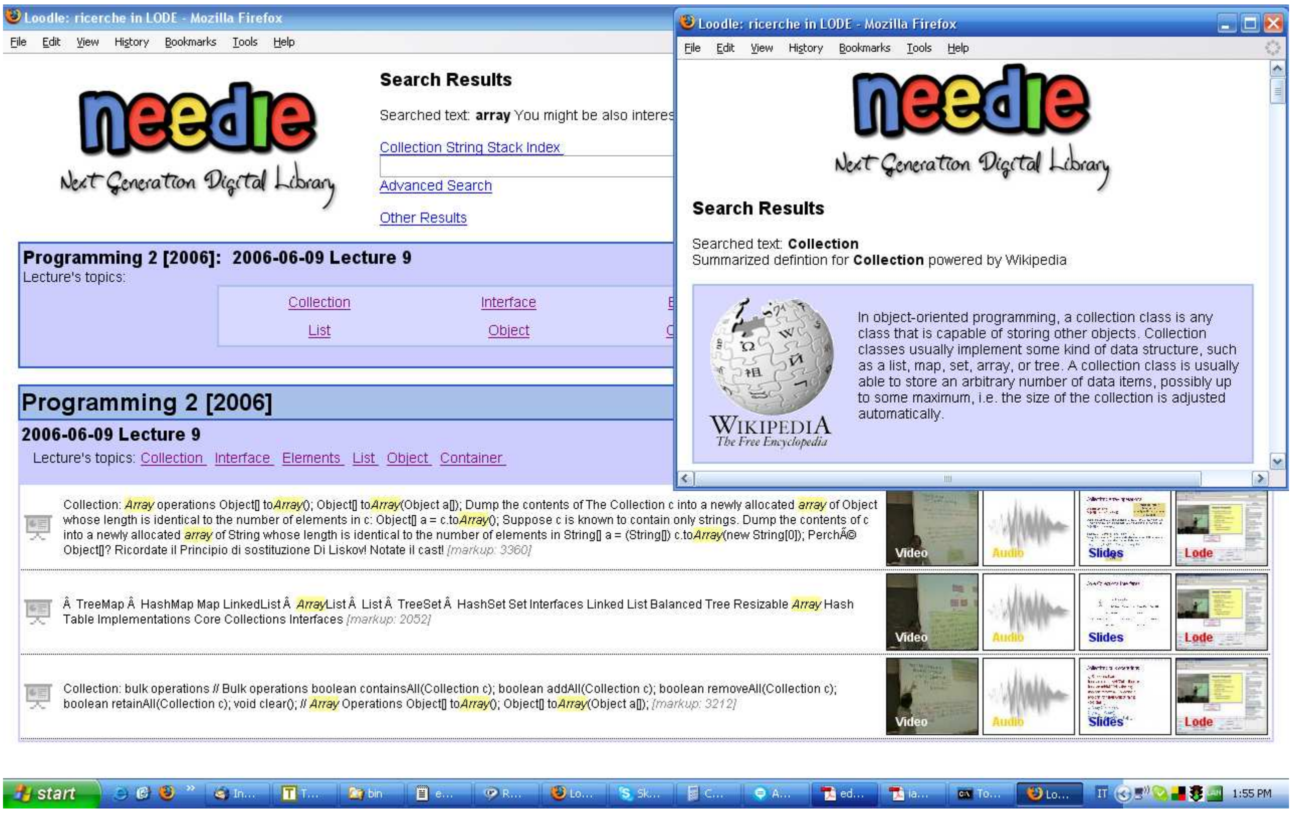
We can summarize with the following points the features where our approach could contribute:

- *Search Suggestion and Query Expansion*
  Wikipedia is used for finding topics related to the searched one. In our search user interface we show the hits for the searched string and a bunch of links to some related topics which have a correspondence in our repository. A click on one of the link will initiate a search for the occurrences of the link term in the learning material. This is done by viewing all the strong links retrieved through Wikipedia which term appear also in the event material, in this way we can suggest different search terms or topics that are connected to the first searched one.
- Automatic Annotation
  For each occurrence displayed in the hits, we show some links to related important topics. The important topics automatically annotate the event with some terms which have a predefined meaning in Wikipedia.

In this way there are no more ambiguities in the meaning of a term used for annotation. Another advantage of the strong link identification in combination with the term vector extracted for every event is the possibility to automatically describe the most important concepts of the event.

- Automatic Summarization
  The semantic discovery approach described in the previous section brought us to the individuation of the strong links for each topic. Based on them we can generate for each event annotation (topic) a brief summarization of the description of the topic in Wikipedia. A click on one of the event annotation will display the summary plus the retrieved hits for that term. In our search user interface for each event(lecture, seminar, meeting) we show the six most important words and the related summarized Wikipedia definitions.

In figure 5.1 the possible features derived by the implementation of the semantic discovery approach described above are shown. Figure 5.1 is a screen shot of NEEDLE where every lecture it is first summarized by means of a list of important topics. So the user looking at the important topics list can understand if a lecture in the hits is relevant or not for each search and in case s/he can go in depth looking at the details hits inside the lecture itself. Every hit in the lecture is composed of a brief textual description an four presentation modalities. The result can be analyzed watching part of the video where the hit have been found, listening only to the audio, only watching the associated slide or having a combine view, where video and slide are time-synchronized.


Loodle: ricerche in LODE - Mozilla Firefox
File Edit View History Bookmarks Tools Help
needle
Next Generation Digital Library
Video
Audio
Slides
Lode
start
1:55 PM

Search Results

Searched text: **array** You might be also interes

Collection String Stack Index

Advanced Search

Other Results

Programming 2 [2006]: 2006-06-09 Lecture 9
Lecture's topics:

Collection Interface
List Object

Programming 2 [2006]

2006-06-09 Lecture 9
Lecture's topics: Collection Interface Elements List Object Container

Collection: *Array* operations Object[] to*Array*(); Object[] to*Array*(Object a[]); Dump the contents of The Collection c into a newly allocated *array* of Object whose length is identical to the number of elements in c: Object[] a = c.to*Array*(); Suppose c is known to contain only strings. Dump the contents of c into a newly allocated *array* of String whose length is identical to the number of elements in String[] a = (String[]) c.to*Array*(new String[0]); PerchÃ© Object[]? Ricordate il Principio di sostituzione Di Liskov! Notate il cast! *[markup: 3360]*

Â TreeMap Â HashMap Map LinkedList Â *Array*List Â List Â TreeSet Â HashSet Set Interfaces Linked List Balanced Tree Resizable *Array* Hash Table Implementations Core Collections Interfaces *[markup: 2052]*

Collection: bulk operations // Bulk operations boolean containsAll(Collection c); boolean addAll(Collection c); boolean removeAll(Collection c); boolean retainAll(Collection c); void clear(); // *Array* Operations Object[] to*Array*(); Object[] to*Array*(Object a[]); *[markup: 3212]*

Search Results

Searched text: **Collection**
Summarized defintion for **Collection** powered by Wikipedia

In object-oriented programming, a collection class is any class that is capable of storing other objects. Collection classes usually implement some kind of data structure, such as a list, map, set, array, or tree. A collection class is usually able to store an arbitrary number of data items, possibly up to some maximum, i.e. the size of the collection is adjusted automatically.

Fig. 5.1. *An application of semantics extraction through Wikipedia*

On the right most part of figure 5.1 a pop-up window for with a description of one of the important terms of the lecture can be viewed by clicking on the term. The description is actually the summary of the Wikipedia article which refers to the term itself. In this way there is no ambiguity with the meaning of a term used for describing the multimedia content. Below the input field designed for running new searches there is the implementation of the related topics feature. Based on the last user search, the system advices the user for related concepts. A click on one related term will initiate a new search. In this way the user can discover new connections between topics.

6. Future directions. One of the activities we plan for the near future is related to extending the Wikipedia module to support various languages. The multilanguage support consists in recognizing relations between terms in the corpus which are not in English. As a first step, wed'll look at the links to the other instances of Wikipedia in different languages. In most cases, pages in the Wikipedia instance in one language have links to pages in many other Wikipedia instances in other languages. Since these links were created manually by the page authors, in most cases there is no ambiguity in the translation. In case a link to the target language of interest is not present, we can resort to freely available, albeit less trustable external sources for translating from and to English. The Wikipedia process described in the previous section will not change but writing language dependent processing modules such as language specific stemmers should be added to enable the comparison of the related Wikipedia content found in English with the terms contained in the multimedia content repository. Consequently we have scheduled an evaluation of the presented approach for annotating a large amount of text resources and a user based evaluation to assess if the introduction of semantic multimedia information retrieval is actually bringing an advantage to the student. We will carry out a studentd's performance evaluation on some topics presented in the e-Learning repository and we will compare the results with the ones we gathered last year using a text based search that was not semantically enhanced [9].

7. Conclusion. In this paper we described an approach to semantically annotate the content of an unstructured multimedia repository. The annotation has been done combining the terms extracted from the corpus with lexicographic relationships from Wikipedia. Wikipedia has been used as an alternative to ontologies. The content annotated in this way permits to keep track of the relations between annotations. The approach has been used for giving search suggestions in multimedia information retrieval, in multimedia annotation and for giving a brief description of the topics of the multimedia event. Our approach is domain independent, and it could in theory also be applied to different use cases where there is a need for clustering or annotation of a corpus.

REFERENCES

[1] K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, eds., *DBpedia: A Nucleus for a Web of Open Data*, vol. 4825 of Lecture Notes in Computer Science, Springer, 2007.

[2] S. Auer and J. Lehmann, *What have innsbruck and leipzig in common? extracting semantics from wiki content.*, in Proceedings of the 4th European Semantic Web Conference, ESWC 2007, 2007.

[3] M. Bertini, A. D. Bimbo, and C. Torniai, *Enhanced ontologies for video annotation and retrieval*, in MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, New York, NY, USA, 2005, ACM, pp. 89–96.

[4] K. Bontcheva, D. Maynard, H. Cunningham, and H. Saggion, *Using human language technology for automatic annotation and indexing of digital library content.*, in 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'2002), Rome, September 2002.

[5] M. Dolzani and M. Ronchetti, *Video streaming over the internet to support learning: the lode system*, in WIT Transactions on Informatics and Communication Technologies, vol. 34, 2005, pp. 61–65.

[6] M. Dowman, V. Tablan, H. Cunningham, and B. Popov, *Web-assisted annotation, semantic indexing and search of television and radio news*, in Proceedings of the 14th International World Wide Web Conference, Chiba, Japan, 2005.

[7] A. Ebersbach, M. Glaser, and R. Heigl, *Wiki : Web Collaboration*, Springer, November 2005.

[8] A. Fogarolli, G. Riccardi, and M. Ronchetti, *Searching information in a collection of video-lectures*, in Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, Vancouver, Canada, June 2007, AACE, pp. 1450–1459.

[9] A. Fogarolli and M. Ronchetti, *Case study: Evaluation of a tool for searching inside a collection of multimodal e-lectures*, in Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, Vancouver, Canada, June 2007, AACE, pp. 3893–3900.

[10] M. Hepp, K. Siorpaes, and D. Bachlechner, *Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management.*, Internet Computing, 11 (2007), pp. 54–65.

[11] L. Obrst, *Ontologies for semantically interoperable systems*, in CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management, New York, NY, USA, 2003, ACM Press, pp. 366–369.

[12] S. Ponzetto and M. Strube, *Deriving a large scale taxonomy from wikipedia*, in Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07), Vancouver, B.C., July 2007.

[13] J. Porter, *Folksonomies: A user-driven approach to organizing content*, User Interface Engineering, (2005).

[14] F. Suchanek, G. Kasneci, and G. Weikum, *Yago: A large ontology from wikipedia and wordnet*, Research Report MPI-I-2007-5-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, 2007.

[15] J. T. Tennis, *Social tagging and the next steps for indexing*, in 17th SIG/CR Classification Research Workshop, 2006.

[16] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer, *Semantic wikipedia*, in Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006, MAY 2006.

[17] N. Weber and P. Buitelaar, *Web-based ontology learning with isolde*, in Proc. of ISWC2006 Workshop on Web Content Mining with Human Language Technologies, 2006.

[18] C. Welty and J. Murdock, *Towards knowledge acquisition from information extraction*, The Semantic Web - ISWC 2006, (2006), pp. 709–722.

[19] F. Wu and D. Weld, *Autonomously semantifying wikipedia*, in ACM Sixteenth Conference on Information and Knowledge Management (CIKM-07), Lisbon, Portugal, November 2007.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: November 23rd, 2007
Accepted: January 15th, 2008
Extended version received: June 29th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 271–279. http://www.scpe.org

ISSN 1895-1767
© 2008 SCPE

UNEVENNESS IN NETWORK PROPERTIES ON THE SOCIAL SEMANTIC WEB

RAF GUNS*

Abstract. This paper studies unevenness in network properties on the social Semantic Web. First, we propose a two-step methodology for processing and analyzing social network data from the Semantic Web. Using the SPARQL query language, a derived RDF graph can be constructed that is tailored to a specific question. After a brief introduction to the notion of unevenness, this methodology is applied to examine unevenness in network properties of semantic data. Comparing Lorenz curves for different centrality measures, it is shown how examinations of unevenness can provide crucial hints regarding the topology of (social) Semantic Web data.

Key words: semantic Web, social network analysis, SPARQL, unevenness

1. Introduction. The *social Semantic Web* is a broad, non-technical term, referring to data on the Semantic Web (encoded in RDF) that contain social information. The most prevalent ontology on the social Semantic Web is the FOAF (Friend Of A Friend) vocabulary [9]. FOAF can express information "about people and the things they make and do" and especially about how they are related. In this article, we will use a socio-cultural ontology that is (partly) based on FOAF and also uses concepts from other well-known ontologies like Dublin Core.

The Semantic Web [5] in general is conceived as a large-scale distributed information system. While some constituents are still in development and its current uptake is relatively modest, the Semantic Web graph already shows the traits of a complex system. Complex systems are encountered in many different contexts and include such diverse examples as computer networks, social networks, neural networks and cellular networks [13]. As a complex system, the Semantic Web is characterized by [3, 17]:

- *Small world properties*: Made famous by Stanley Milgram's [25] letter experiment, the small world notion refers to the fact that the average shortest path length in a graph is very short (comparable to that of a random graph). In practice, this means that it takes only a few steps to reach any other (reachable) node in the network. It is advisable to also take the longest shortest path, known as the *diameter*, into account. During the last decade, several models have been proposed to account for the small-world effect [26, 31].
- *High clustering*: The neighbours of a given node are likely also neighbours of each other.
- *Skewed degree distribution*: The probability $P(k)$ that a node has degree k (is connected to k other nodes) is not randomly distributed. Instead, it follows a power law $P(k) \approx Ak^{-\gamma}$. Moreover, complex systems typically exhibit power law distributions in more than one way. With regard to the Semantic Web, previous research has shown that a diversity of relations—such as the relation between websites (domain names) and their number of Semantic Web documents or the relation between an ontology and its frequency of use—follows a power law [15].

These properties, however, raise several questions as well. In this article, we first discuss a two-step methodology for extracting the Semantic Web data (or 'semantic data' for short) that we are interested in from the rest. We then focus on the last characteristic and try to compare the skewedness of several network measures. We try to provide an answer to the following two research questions.

First, how can data on the social Semantic Web be used for Social Network Analysis (SNA)? Significant research in this area has already been performed by, among others, Ding et al. [15] and Peter Mika [23, 24]. Much work has concentrated on acquiring and aggregating data (often FOAF data),—especially merging information about unique persons turns out to be far from trivial. In the present article, we assume that 'clean' semantic data are already available and concentrate on the following step: the development of a methodology for using one single RDF graph as the 'master', which can be used as the basis for several kinds of SNA. Ideally, we want to keep as much information as possible and extract a multitude of potentially interesting relations. This particular aspect has received less attention so far.

Second, it is very rarely examined how skewed a distribution is. How can this notion be measured? Quantification of unevenness is crucial for a thorough understanding of a power law distribution; moreover, it can be used for comparison purposes between distributions and between networks.

[*] University of Antwerp, Informatie—en Bibliotheekwetenschap, CST, Venusstraat 35, 2000 Antwerpen, Belgium, raf.guns@ua.ac.be

Both questions will be discussed and demonstrated using semantic data from Agrippa. Agrippa is the catalogue and database of the Archive and Museum of Flemish Cultural Life (AMVC Letterenhuis, located in Antwerp, Belgium). Where applicable, the RDF version builds upon existing ontologies like FOAF and Dublin Core. Agrippa contains a wealth of information about both the archived materials and the socio-cultural actors (people and organizations) that have created them. We will mostly use Agrippa information about the 237,062 letters present at the AMVC Letterenhuis and their writers and recipients.

2. Two-step methodology. Semantic data can be stored in many different ways: as a (set of) document(s) in one of the many RDF syntaxes [4]; in a 'classic' relational database; or in a triplestore, a dedicated RDF database. For performance and convenience reasons, we are using a triplestore, but most techniques can also be performed on, for instance, RDF documents. The triplestore used is Sesame, freely available at `http://www.openrdf.org/`.[1]

Partly due to their distributed nature, semantic data may appear quite dazzling: many different kinds of data, drawn from several ontologies, between which a multitude of relations exist. How can one make heads or tails out of them? Assuming the existence of a set of fairly clearly defined questions to be answered, we propose a two-step methodology, which critically depends on the SPARQL query language [27] or a query language with similar capabilities. In short, the two steps are:

1. Construct an extraction query in SPARQL and apply it to the RDF graph. This yields a derived graph, specifically tailored to the question(s).
2. Convert the derived graph to a format intended for SNA.

We will now discuss both steps in greater detail, using a part of Agrippa as an example (shown in Figure 2.1). Both Organization and Person are a kind of Agent. A LetterContext ties together the different participants in the act of letter-writing: the writer(s), the recipient(s) and the letter as a physical object. A letter can be written and received by either an Agent or an AffiliationContext. This refers to a person (the 'affiliatee') acting on behalf of his/her affiliation to an organization (the 'affiliator').

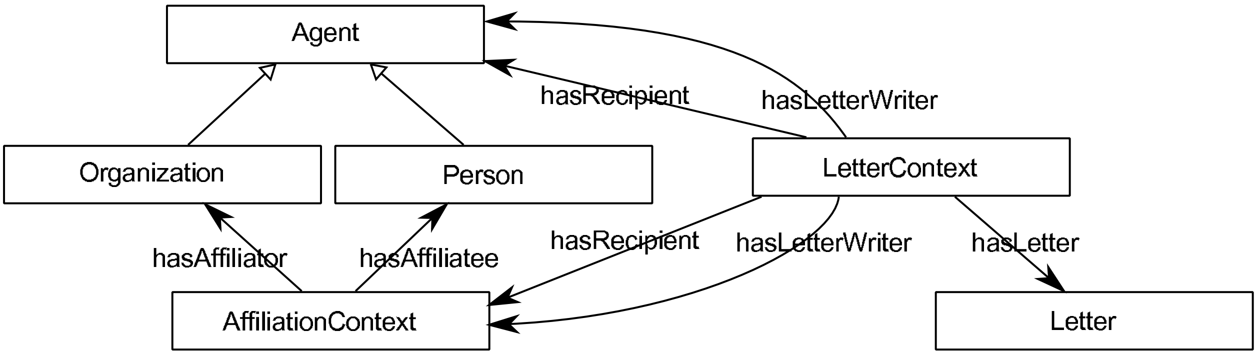


FIG. 2.1. *Part of the Agrippa ontology, showing the relations between six classes*

2.1. SPARQL information extraction. Four SPARQL query types exist: `SELECT`, `CONSTRUCT`, `ASK` and `DESCRIBE`. SPARQL queries are usually `SELECT` queries, which return a table of results. In this step, we employ `CONSTRUCT` queries, which return a new RDF graph. A similar architecture can also be found in the MESUR project [8, 28]. We will refer to the original graph as *source graph* and to the newly constructed graph as *derived graph*.

First, we compare the original graph in the triplestore and the questions to be answered. Some questions simply involve the extraction of parts of the RDF graph (ignoring the rest), like the following example. Suppose we want to examine only those letters that were created in an organizational context. This boils down to extracting the letters that are written by an Organization or an AffiliationContext:

```
PREFIX : <http://anet.ua.ac.be/agrippa#>
CONSTRUCT {
  ?context a                :LetterContext ;
```

[1] For an overview of triplestores, see [20].

```
           :hasLetterWriter ?writer ;
           :hasRecipient    ?recipient ;
           :hasLetter       ?letter .
}
WHERE {
  ?context a                :LetterContext ;
           :hasLetterWriter ?writer ;
           :hasRecipient    ?recipient ;
           :hasLetter       ?letter .
  { ?writer a :Organization } UNION
  { ?writer a :AffiliationContext }
}
```

Other questions also require knowledge on how relations in the model interact,—these involve both extraction and combination of parts of the model. Here are two examples from Agrippa. The following query constructs a derived graph of persons and their affiliations to organizations. The result is a bipartite graph, i. e. a graph with two kinds of nodes (persons and organizations).

```
PREFIX : <http://anet.ua.ac.be/agrippa#>
CONSTRUCT { ?person :affiliatedWith ?org }
WHERE {
  ?aff :hasAffiliator ?org ;
       :hasAffiliatee ?person .
}
```

And the following query constructs a simple derived graph that links author(s) and recipient(s) of each letter:

```
PREFIX : <http://anet.ua.ac.be/agrippa#>
CONSTRUCT { ?sender <urn:agrext#writesLetterTo> ?recipient }
WHERE {
  ?context :hasLetterWriter ?sender ;
           :hasRecipient    ?recipient .
}
```

It should be noted that it is often easier to obtain the desired results using one or more intermediate extraction queries. As such, a derived graph may become the source graph in a next step and so on. One could, for example, use the result of the first example as the source graph for the third example query. Although extraction queries are obviously not as powerful as a dedicated program or full-fledged reasoner, they are often sufficient and much faster to implement.

One of the advantages of storage in a triplestore is availability of the SPARQL protocol [14]. As its name implies, the SPARQL protocol is designed for exchanging SPARQL queries and results between clients and servers. It is entirely based on Web standards like HTTP and XML.

2.2. Conversion for SNA analysis. Once a derived graph has been obtained, it can be studied. There exist several projects for visualizing and exploring RDF and FOAF data, such as FOAF Explorer,[2] RDF-Gravity[3] and Visual Browser.[4] These tools, however, generally do not provide SNA measures like centrality and clustering, although Flink [23] seems a promising exception. Moreover, they generally do not scale to very large graphs. As long as there exist virtually no applications that successfully bring network analysis to RDF, it seems advisable to convert the derived graph to a more generic file format for network analysis.

Thus, while not strictly necessary, this step ensures compatibility with other SNA efforts and permits techniques that are difficult to perform on plain RDF graphs. We handle these conversions by integrating with pyNetConv, a Python library that can convert to most common formats, including Pajek, NetworkX, and GML.

[2] http://xml.mfd-consult.dk/foaf/explorer/

[3] http://semweb.salzburgresearch.at/apps/rdf-gravity/

[4] http://nlp.fi.muni.cz/projekty/visualbrowser/

3. Unevenness.

3.1. The Lorenz curve and the Gini evenness index. The distribution of degrees on the Semantic Web is—like many other relations—highly uneven: a small number of nodes has a huge amount of links, while the vast majority has very few. How can this unevenness be quantified?

Unevenness or inequality has been studied extensively in econometrics and informetrics. Since not all existing measures satisfy all necessary requirements [1, 16], we will limit the present discussion to two methods, using the following simple array as an example: $X = (1, 3, 4, 7, 10, 15)$. These numbers could express the distribution of wealth, the number of publications per author or the number of links per node. Clearly, there is some unevenness, but how much exactly?

The Lorenz curve [21] is a graphical representation of unevenness. First, we determine the relative amounts:

$$a_i = \frac{x_i}{\sum x}$$

resulting in (1/40, 3/40, 1/10, 7/40, 1/4, 3/8). The horizontal axis of the Lorenz curve has the points i/N ($i = 1, 2, \ldots, N$). The vertical axis of the Lorenz curve has their cumulative fraction: $a_1 + a_2 + \ldots + a_i$. We thus construct the Lorenz curve (Figure 3.1). The diagonal line represents the case of perfect evenness—everyone possesses the same amount. The further the curve is removed from the diagonal, the greater the unevenness. Note that we have ranked our numbers in increasing order, resulting in a convex Lorenz curve. The concave Lorenz curve results from ranking in decreasing order and is completely equivalent. Complete unevenness—one person has everything, and the rest nothing—would be represented as a convex curve following the bottom and the right side of the plot.

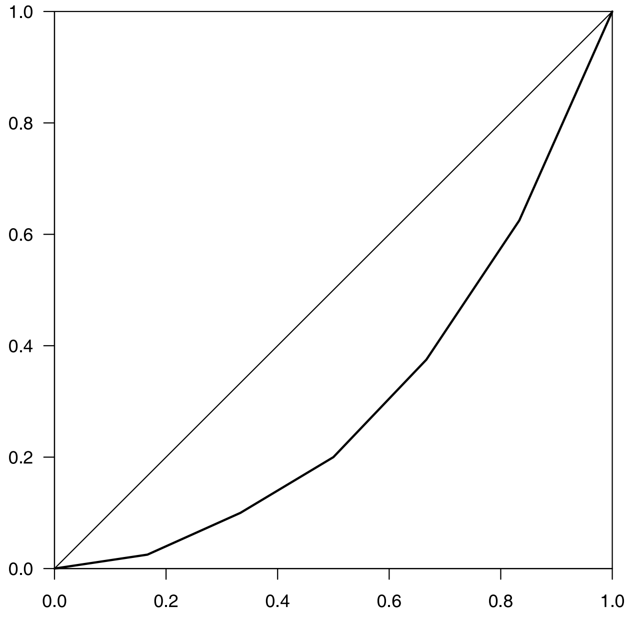


FIG. 3.1. *Convex Lorenz curve of the array (1, 3, 4, 7, 10, 15)*

Suppose we want to express this unevenness in a number. A good measure is the Gini evenness index G' [29], originally devised to characterize the distribution of wealth over social classes [18],

$$G'(X) = \frac{2}{\mu N^2} \left(\sum_{j=1}^{N} (N + 1 - j) x_j \right) - \frac{1}{N}$$

with x_j ranked in increasing order and μ the mean of the set x_j. There exists a direct relation between the Lorenz curve and the Gini evenness index: G' is equal to twice the area under the convex Lorenz curve.

Lorenz curves determine a partial order: in some, but not all, cases, an order can be determined from the comparison of two Lorenz curves. Indeed, if one convex Lorenz curve is completely below another, then the former expresses less evenness than the latter. It should be stressed that Lorenz curves may 'overlap' or cross each other. In these cases, no order can be determined from the curves [29].

3.2. Application to Agrippa.

3.2.1. Overview of network measures. Let us take the author-recipient graph constructed in the last example of 2.1 $N = 40,914$ as an example. Each node is connected by 5.08 links on average, but the actual in- and out-degree follow a power law distribution (Figure 3.2). We will consider the following network measures, most of which are defined by Wasserman & Faust [30]:

- *Degree centrality (DC)*: is the number of links connected to a given node.
- *Betweenness centrality (BTC)*: characterizes the importance of a given node for establishing short pathways between other nodes.
- *Closeness centrality (CC)*: characterizes how fast other nodes can be reached from a given node.
- *Pagerank (PR)*: characterizes the importance of a given node by combining its number of in-links with the importance of the nodes that link to it. The algorithm was originally created for determining a web page's importance [10] but has since been used in many other contexts as well (e.g., [12, 22]).

This small list of measures is in no way intended to be exhaustive. Many other measures exist and even the ones listed here have several varieties themselves. They have been chosen because they are both well-known and generally used and accepted. Moreover, they can be computed using standard software tools. For the current article, we used the *igraph* R package, available at `http://cneurocvs.rmki.kfki.hu/igraph/`.

The centrality measures listed above all have variants for directed and undirected networks, but we will only consider the directed variants. Both degree centrality and closeness centrality have different algorithms for in-links and out-links. We can distinguish between in-degree centrality (IDC) and out-degree centrality (ODC), and between in-closeness centrality (ICC) and out-closeness centrality (OCC). This distinction is not useful for betweenness centrality and PageRank.

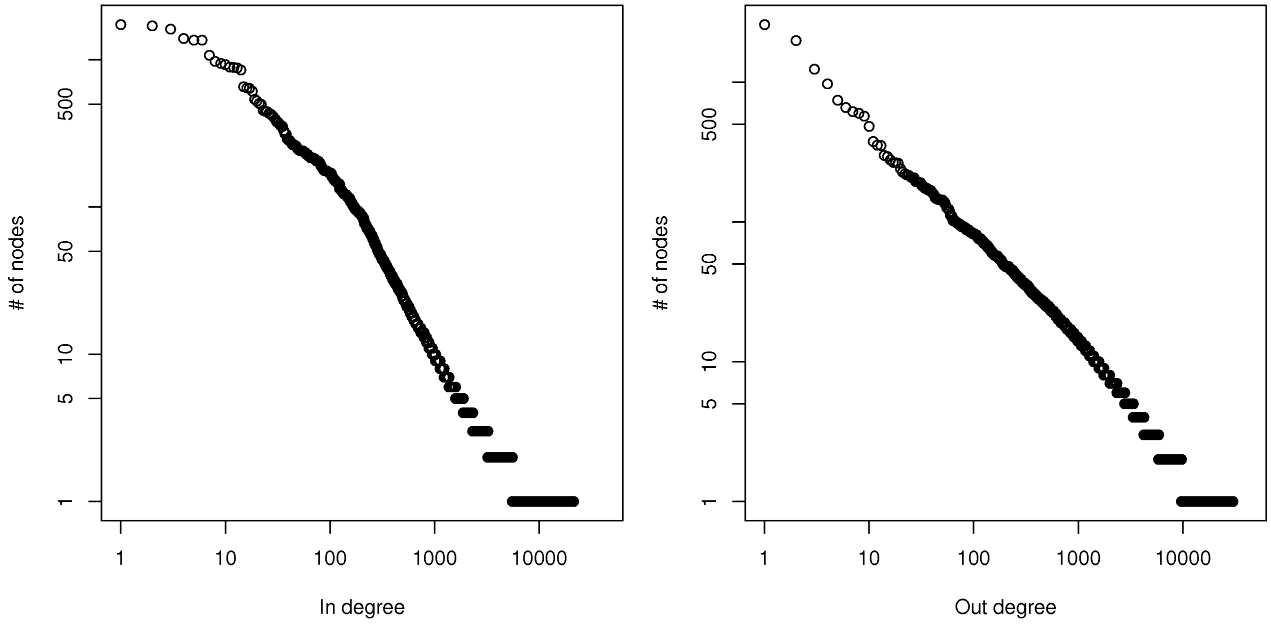


FIG. 3.2. *Power law distribution for in-degree and out-degree*

3.2.2. Comparison of unevenness between network measures. The graph is not fully connected, but the main component ($N = 40,303$) accounts for the vast majority of nodes (98.5%). Henceforth, we will only consider the nodes that are part of the main component, since very small components (e.g., $N = 2$) can distort the overall picture. For instance, a node v in such a component may have $CC_v = 1$, even if its position in the overall network is obviously marginal. We therefore consider it methodologically more correct to only consider nodes that are part of the main component.

Comparing IDC to ODC and ICC to OCC (Figure 3.3), we see that in both cases the measure based on in-links is more uneven. In spite of this difference, it should be noted that in both cases the shape of the Lorenz curve of the in-link-based measure is similar to that of the out-link-based one.

PageRank is, in a sense, a more refined version of in-degree centrality. Whereas the latter only considers the local neighbourhood (i. e. the number of links to a given node), PageRank also considers the status of the nodes that are linking to a given node by iteratively passing status between nodes. Figure 3.4 shows that PageRank is actually more even than in-degree centrality. In other words: some extreme variations in degree are 'evened out' by looking at a node's status in the entire network rather than just its number of in-links. Inspection of the data reveals that this is almost exclusively due to nodes with a low number of in-links from some very high status nodes. Put another way, differences between PageRank and IDC may be due to IDC either 'overrating' or 'underrating' some nodes; at least for this example, the latter is mostly the case. Despite the outliers, PageRank and in-degree centrality are highly correlated. Figure 3.4 also illustrates the usefulness of the Lorenz curve for comparing different measures: it makes it possible to, for instance, compare raw numbers (IDC) to normalized ones (PageRank).

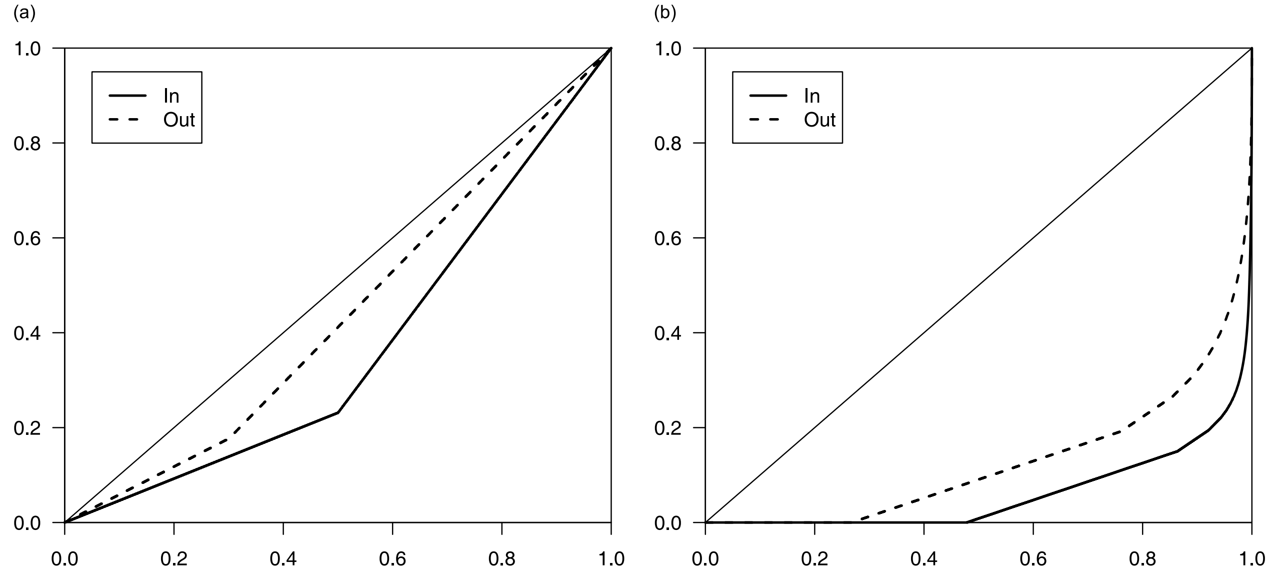


Fig. 3.3. *Comparison of unevenness between in-link-based and out-link-based measures. (a) Comparison of ICC to OCC, (b) Comparison of IDC to ODC*

Betweenness centrality is remarkably uneven (Figure 3.5). Indeed, we immediately see that more than 80% of all nodes have zero betweenness centrality. The Lorenz curve clearly reveals that betweenness centrality is considerably less even than any of the other measures discussed here.

3.3. Discussion. Comparing the Lorenz curves of the different centrality measures reveals a remarkably diversified picture. Betweenness centrality is clearly least even of all. Subsequently, we get degree centrality, PageRank and closeness centrality. The Gini evenness indices basically tell the same story and are summarized in Table 3.1.

As a tentative explanation, we suggest that these differences may be largely due to the small-world effect [26, 31]. Even marginal nodes are relatively close to all others, accounting for minimal differences in closeness. Indeed, the length of the diameter—the longest shortest path—is only 11 and the average shortest path length only 4.12!

As a whole, the graph fits well into the bow-tie or corona models [6, 7, 11], which were originally devised for modelling and explaining link structure on the World Wide Web. The core of the main component is the Largest Strongly Connected Component or LSCC ($N = 9,723$), a component in which any node can be reached (obeying the direction of the links). The LSCC itself has a nucleus of hubs [13, 19], through which almost all other shortest paths pass. These hub nodes typically have extremely high degree centrality. This has two

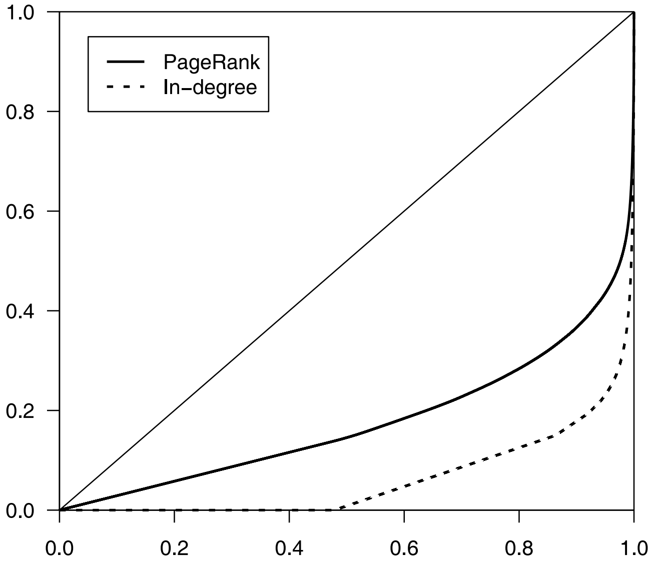


FIG. 3.4. *Comparison of unevenness between PageRank and in-degree centrality*

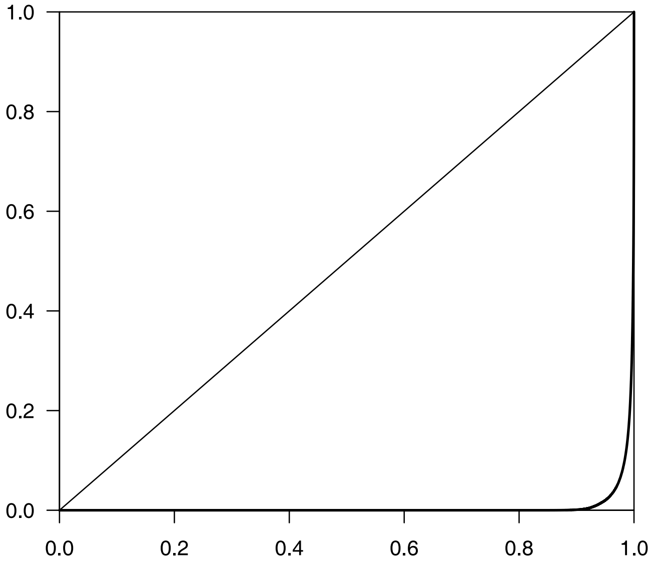


FIG. 3.5. *Unevenness of betweenness centrality*

interesting, seemingly opposite, effects. On the one hand, closeness is increased and closeness centrality becomes more even. On the other hand, it brings about a very uneven betweenness centrality distribution.

PageRank distribution is more even than one might intuitively expect. The hubs have a high status, which is partially transmitted to each of the nodes they link to. As such, a large number of nodes gains a higher PageRank than might be expected from their in-degree centrality or betweenness centrality. Indeed, even if no shortest paths pass through them, their PageRank will still be relatively high. This property of PageRank is very desirable for ranking Web pages, but may be unwanted in some applications of SNA.

4. Conclusions. We have shown how SPARQL can be used in processing social Semantic Web data in a simple two-step methodology, converting the source graph to a better suited derived graph. While SPARQL is obviously less powerful than a 'real' reasoning engine or a dedicated program, it is often sufficient and may well prove simpler and faster to implement. RDF tools are generally not geared towards SNA, although Flink [23] incorporates some basic SNA statistics. Therefore, conversion to other formats is currently recommendable but, luckily, straightforward.

Table 3.1
Gini evenness index of all centrality measures in increasing order of evenness

Centrality measure	G'
Betweenness centrality	0.01
In-degree centrality	0.12
Degree centrality (in and out)	0.25
Out-degree centrality	0.26
PageRank	0.35
In-closeness centrality	0.73
Out-closeness centrality	0.88
Closeness centrality (in and out)	0.94

The Lorenz curve and the Gini evenness index G' are two excellent methods for studying unevenness. Taking Agrippa as a concrete example, it can be seen that unevenness measures may confirm or enforce hypotheses regarding the network topology. In the example discussed, the massive difference between betweenness centrality and closeness centrality distribution confirms the small-world hypothesis and reveals the topology of the graph with a small nucleus, through which most other paths must pass. The example also illustrates the need for a wide variety of centrality measures: they are indeed very different (as is obvious from just comparing the Lorenz curves) and each reveals a different aspect of the network.

Most of these results, such as the establishment of the small-world effect, could have been achieved without studying the unevenness of network properties. Consequently, the current paper should be regarded as a first step: it illustrates how unevenness measures can be used to achieve results similar to existing, well-established methods. In future research, we hope to expand upon these results by studying a greater variety of (social) networks, including different classes of small-world networks [2].

Acknowledgements. A previous version of this paper was presented at the Social Aspects of the Web 2008 Workshop, held in conjunction with the BIS2008 conference in Innsbruck. I thank prof. Ronald Rousseau and the organizers and participants to the Workshop for their constructive comments, and prof. Richard Philips for providing access to the Agrippa dataset.

REFERENCES

[1] P. D. Allison, *Measures of inequality*, American Sociological Review, 43 (1978), pp. 865–880.

[2] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, *Classes of small-world networks*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 11149–11152.

[3] D. Bachlechner and T. Strang, *Is the semantic web a small world?*, in Proceedings of the Second International Conference on Internet Technologies and Applications (ITA 07), 09 2007.

[4] D. Becket, *New syntaxes for rdf.* `http://www.dajobe.org/2003/11/new-syntaxes-rdf/paper.html` 2004.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, *The Semantic Web*, Scientific American, 284 (2001), pp. 28–37.

[6] L. Björneborn, *Small-World Link Structures across an Academic Web Space: A Library and Information Science Approach*, PhD thesis, Royal School of Library and Information Science, Denmark, 2004.

[7] ———, *'mini small worlds' of shortest link paths crossing domain boundaries in an academic web space*, Scientometrics, 68 (2006), pp. 395–414.

[8] J. Bollen, M. A. Rodriguez, H. Van de Sompel, L. Balakireva, W. Zhao, and A. Hagberg, *The largest scholarly semantic network... ever.*, in Proceedings of the 16th international conference on World Wide Web, ACM Press New York, NY, USA, 2007, pp. 1247–1248.

[9] D. Brickley and L. Miller, *Foaf vocabulary specification 0.91. namespace document 2 november 2007 ij openid edition.* http://xmlns.com/foaf/spec/, 2007.

[10] S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Computer Networks and ISDN Systems, 30 (1998), pp. 107–117.

[11] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, *Graph structure in the web*, Computer Networks, 33 (2000), pp. 309–320.

[12] P. Chen, H. Xie, S. Maslov, and S. Redner, *Finding scientific gems with google's pagerank algorithm*, Journal of Informetrics, 1 (2007), pp. 8–15.

[13] C. Christensen and R. Albert, *Using graph concepts to understand the organization of complex systems*, Nov 2006.

[14] K. G. Clark, L. Feigenbaum, and E. Torres, *Sparql protocol for rdf w3c recommendation 15 january 2008.* `http://www.w3.org/TR/rdf-sparql-protocol/` 2008.

[15] L. Ding and T. Finin, *Characterizing the semantic web on the web*, in Proceedings of the 5th International Semantic Web Conference, 2006.

[16] L. Egghe and R. Rousseau, *Transfer principles and a classification of concentration measures*, Journal of the American Society for Information Science, 42 (1991), pp. 479–489.

[17] R. Gil and R. García, *Measuring the semantic web*, in Advances in Metadata Research, Proceedings of MTSR'05, 2006.

[18] C. Gini, *Il diverso accrescimento delle classi sociali e la concentrazione della richezza*, Giornale degli Economisti, 11 (1909).

[19] J. M. Kleinberg, *Authoritative sources in a hyperlinked environment*, in SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 1998, pp. 668–677.

[20] R. Lee, *Scalability report on triple store applications.* `http://simile.mit.edu/reports/stores/` 2004.

[21] M. O. Lorenz, *Methods of measuring the concentration of wealth*, Publications of the American Statistical Association, 9 (1905), pp. 209–219.

[22] R. Mihalcea, P. Tarau, and E. Figa, *Pagerank on semantic networks, with application to word sense disambiguation*, in COLING '04: Proceedings of the 20th international conference on Computational Linguistics, Morristown, NJ, USA, 2004, Association for Computational Linguistics, p. 1126.

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

[24] P. Mika, T. Elfring, and P. Groenewegen, *Application of semantic technology for social network analysis in the sciences*, Scientometrics, 68 (2006), pp. 3–27.

[25] S. Milgram, *The small world problem*, Psychology Today, 2 (1967), pp. 60–67.

[26] M. E. J. Newman, *Models of the small world*, Journal of Statistical Physics, 101 (2000), pp. 819–841.

[27] E. Prud'hommeaux and A. Seaborne, *Sparql query language for rdf. w3c recommendation 15 january 2008.* `http://www.w3.org/TR/rdf-sparql-query/` 2008.

[28] M. A. Rodriguez, J. Bollen, and H. Van de Sompel, *A practical ontology for the large-scale modeling of scholarly artifacts and their usage*, in JCDL '07: Proceedings of the 2007 Conference on Digital Libraries, New York, NY, USA, 2007, ACM Press, pp. 278–287.

[29] R. Rousseau, *Lorenz curves determine partial orders for comparing network structures*, in CREEN (Critical events in evolving networks) Workshop, 2007.

[30] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, Structural Analysis in the Social Sciences, Cambridge University Press, 1994.

[31] D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks.*, Nature, 393 (1998), pp. 440–442.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: February 2nd, 2008
Accepted: March 20th, 2008
Extended version received: July 7th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 281–292. http://www.scpe.org

ISSN 1895-1767

DEEP WEB NAVIGATION BY EXAMPLE

YANG WANG AND THOMAS HORNUNG*

Abstract. Large portions of the Web are buried behind user-oriented interfaces, which can only be accessed by filling out forms. To make the therein contained information accessible to automatic processing, one of the major hurdles is to navigate to the actual result page. In this paper we present a framework for navigating these so-called Deep Web sites based on the page-keyword-action paradigm: the system fills out forms with provided input parameters and then submits the form. Afterwards it checks if it has already found a result page by looking for pre-specified keyword patterns in the current page. Based on the outcome either further actions to reach a result page are executed or the resulting URL is returned.

Key words: form analysis, deep Web navigation by page-keyword-actions

1. Introduction. The Web can be classified into two categories with respect to access patterns: the Surface Web and the Deep Web [7]. The Surface Web consists of static and publicly available Web pages, which contain links to other pages and can be represented as a directed graph. This Web graph can be traversed by crawlers (also known as spiders) and the found pages are then traditionally indexed by search engines.

The Deep Web in contrast consists of dynamically generated result pages of numerous databases, which can be queried via a Web form. These pages cannot be reached by following links from other pages and it is therefore challenging to index their content. Figure 1 depicts the general interaction pattern between the user and a Deep Web site. The user fills out the form field with the desired information (1) and the Web form is sent to the server where it is transformed in a database query. In this phase it is possible, that the system needs further user input due to ambiguity in the underlying data, e.g. there might be too many results for a query, and the user has to provide further information on intermediate pages (2). Finally, the Web server has gathered all necessary information to generate the result page and it is delivered to the user (3).

[12] discovered an exponential growth and great subject diversity of these Deep Web sites. Among others they arrived at the following conclusions:

- There are approximately 43.000—96.000 Web-accessible databases,
- The Deep Web is 400—500 times larger than the Surface Web,
- 95% of the available data and information on the Deep Web is freely available.

Taking into account this vast amount of high-quality data, which is geared towards human visitors, it is not surprising that many different research questions are actively pursued in this area at the moment, e.g. vertical search engines [13].

In this paper we present *DNavigator*, a framework to automate the necessary interaction steps to obtain data from Deep Web sites. The idea is to record the user interactions from the initial Web form to the desired result page. These interactions are generalized, where two different phases can be distinguished: the filling out and submission of the frontend form (cf. (1) in Figure 1) and the actions performed on intermediate pages (cf. (2) in Figure 1). Consequently, DNavigator consists of two components: Web form analysis and Deep Web navigation modeling.

This framework has been motivated by the FireSearch project [15], which is geared towards collecting and analyzing Deep Web data at query time. The ultimate extraction and labeling of data from the result page is done with the ViPER extraction system [23]. However, as the framework has been implemented in JavaScript and Java as a Firefox plugin it could be used with minor modifications in other projects, e.g. for a domain-specific Meta Search engine, where the relevant Deep Web sources could be integrated by an interested community, as well.

The paper is structured as follows: we start with a description of the two main components of our framework, namely the analysis of form fields in Section 2 and the navigation model in Section 3. Section 4 deals with the intricacies of implementing our research prototype in the Firefox browser. In Section 5 we present an evaluation of our system and in Section 6 we discuss related work. Finally, we give an outlook on future work in Section 7 and conclude with Section 8.

*Institute of Computer Science, Albert-Ludwigs University Freiburg, Germany, {wangy,hornungt}@informatik.uni-freiburg.de

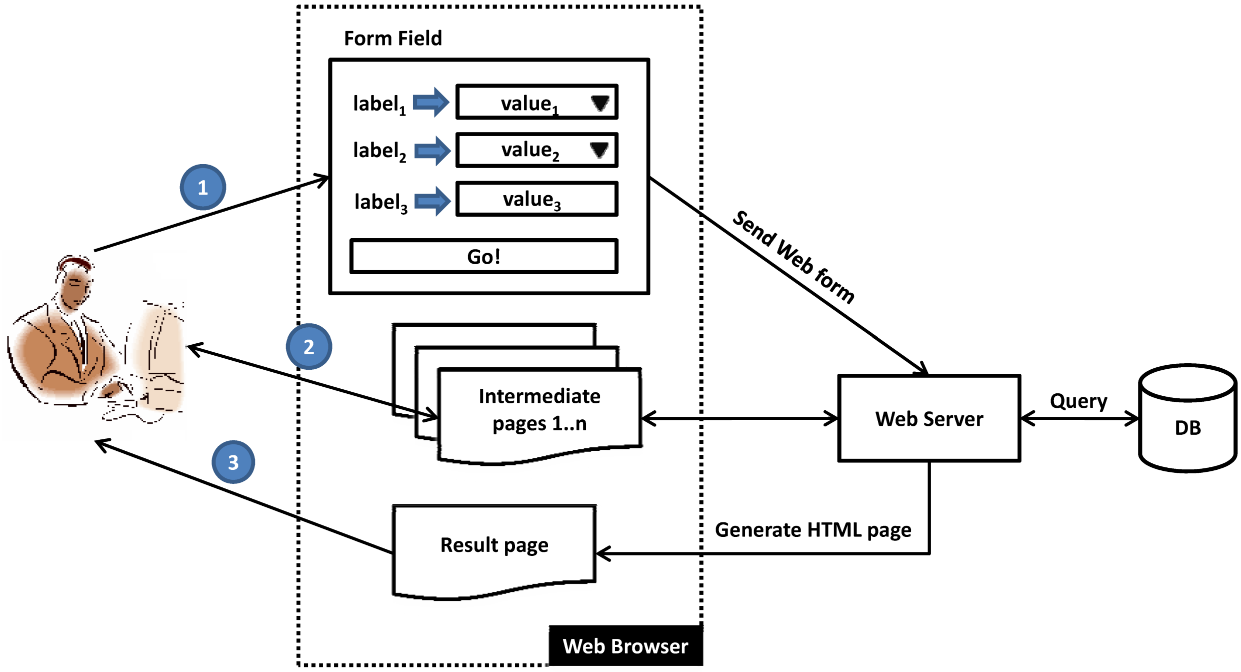
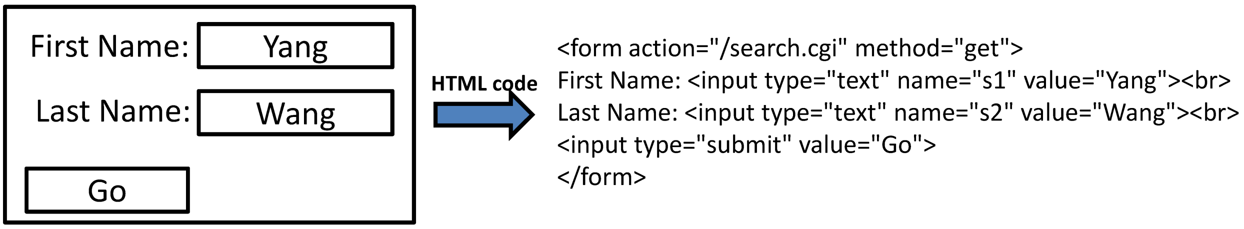


Fig. 1.1. *Accessing a Deep Web site*

2. Form analysis. Web forms are omnipresent: whether the user searches for information on Google, participates in an online vote, or comments an entry in a blog, she always provides information via filling out and submitting a form. On a more technical level, each input element (in the context of this paper we refer to all elements in the form field that can be provided with a value, e.g. checkboxes, as input elements) of a Web form is associated with a unique ID and on submission of the form the value assignments are encoded as either GET or POST HTTP request [3].


First Name: Yang

Last Name: Wang

```
<form action="/search.cgi" method="get">
First Name: <input type="text" name="s1" value="Yang"><br>
Last Name: <input type="text" name="s2" value="Wang"><br>
<input type="submit" value="Go">
</form>
```

Fig. 2.1. *Web form with HTML representation*

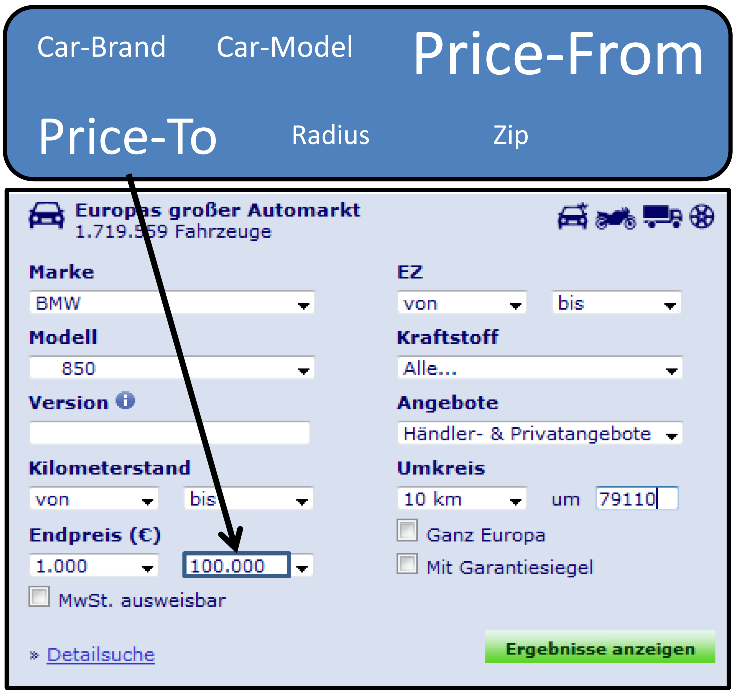
Figure 2.1 shows an example of a simple Web form. The unique ID for the input element labeled *First name* is s1 and thus the associated HTTP GET request looks as follows:

```
 GET /search.cgi?s1=Yang&s2=Wang HTTP/1.1
Host:  www.example.org
User-Agent:  Mozilla/4.0
Accept:  image/gif, image/jpeg, */*
Connection:  close
```

In Section 2.1 we discuss how to map user-defined labels onto input elements, while we deal in Section 2.2 with the problem of dependencies between different input elements. Finally, in Section 2.3 we show how to generate a valid HTTP POST/GET request based on the collected data.

2.1. Labeling of Input Elements. Initially for each new Web page we store all occurring forms with all input elements, IDs and the range of legal values (i. e. for dropdown menu lists, this would be the set of legal options), in a database for later analysis. Afterwards the user can load the desired form field and label the desired input elements, e.g. in Figure 2.2 the maximum desired price the visitor is willing to pay for a used car has been labeled Price-To. The labeling of the Web forms is inspired by the idea of social bookmarking [14]: each user has a personal, evolving vocabulary of tags. Here a tag is a combination of a string label with an XML datatype [15]. The example in Figure 2.2 shows the user vocabulary in the upper corner, where the size of the labels is determined by the frequency they have been used before.

Overall she has labeled six input elements, e.g. the desired brand and the make of the car. Now we check for each labeled input element, if they are static or if there are any dynamic dependencies, which might be due to Ajax interactions with the server. Note, that only these input elements of the form can be used later on for querying that have been labeled in this stage.



Marke: BMW
Modell: 850
Version:
Kilometerstand: von bis
Endpreis (€): 1.000 100.000
□ MwSt. ausweisbar
EZ: von bis
Kraftstoff: Alle...
Angebote: Händler- & Privatangebote
Umkreis: 10 km um 79110
□ Ganz Europa
□ Mit Garantiesiegel

FIG. 2.2. *Labeling of input elements*

Our running example is the analysis of a Web search engine for used cars (`http://www.autoscout24.de`), where each car model depends on its car make. The other input elements are static, i. e. they do not change if one of the other input elements is changing.

2.2. Dependency Check of Input Elements. The dynamic and static combinations are determined automatically after the user has finished labeling the desired input elements based on the following idea: modify the first dropdown menu (only dropdown menus are currently considered as candidates for dynamic elements, all other input element types are assumed to be static by default) and check all other labeled dropdown menus, if the available options have changed. If this is the case, then modify the dependent dropdown menu to uncover layered dependencies and mark the dependent menu as dynamic. After all dropdown menus have been checked, we mark all menus that are not dynamic as static. To avoid loops, we only check possible dropdown menus that have not participated in a dependency in the current analysis cycle before, e.g. in the example shown in Figure 2.2 the car model would not be considered if we check for further dynamic dependencies for the car make input element. Figure 2.3 and Figure 2.4 show the resulting static and dynamic dependencies for our running example.

After the dependency check, the form is submitted and either a POST or GET HTTP request [3] is generated, which encodes the value assignments for the input elements. Here we store the request URL, the action attribute of the form, and the specific value assignments, which are later used for building new requests offline.

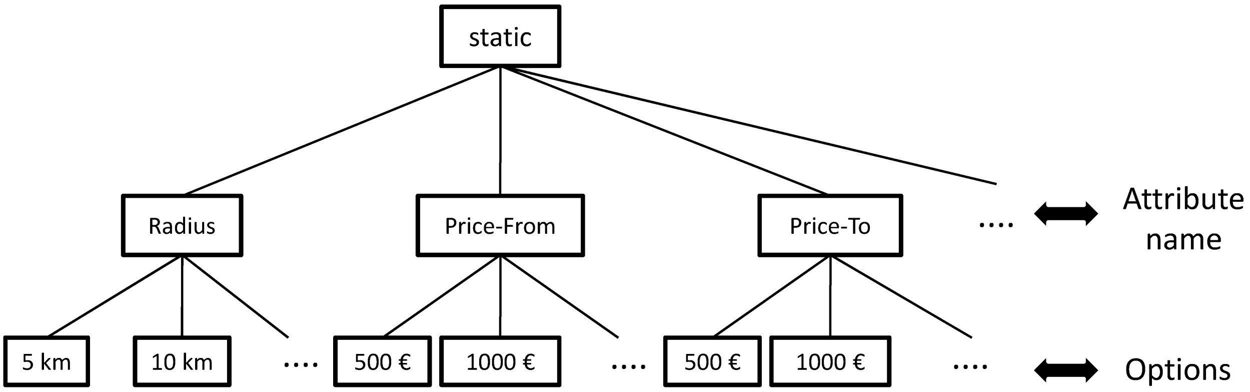


Fig. 2.3. *Relation tree for static input elements for* `http://www.autoscout24.de`

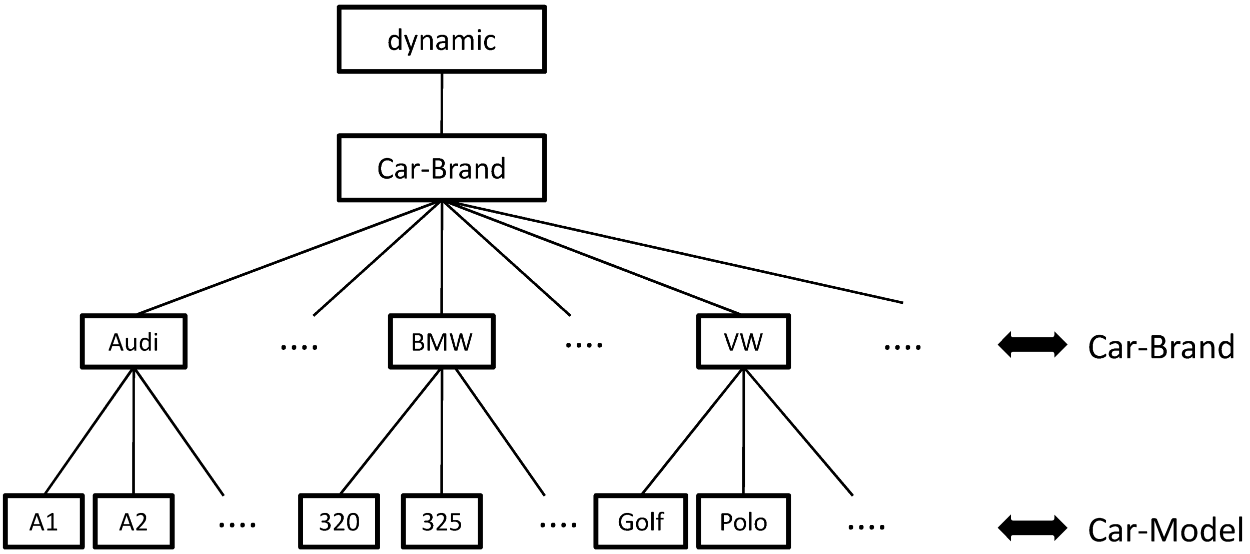


Fig. 2.4. *Relation tree for dynamic input elements for* `http://www.autoscout24.de`

2.3. Simulation of Web Form Behavior. Using the gathered data we now have two possible options to simulate the Web form behavior: we can either use the variable bindings for the user-defined tags to fill out and submit the Web form online, taking into consideration the dynamic and static dependencies or we can directly generate a POST/GET HTTP request offline. For obvious reasons, we usually prefer the offline generation, but as is discussed in Section 5.2 it is sometimes necessary to (automatically) fill out the Web form online.

Suppose the user provided the following variable bindings for our running car search example

```
Car-Brand=BMW
Car-Model=850
```

and the originally captured request URL was

```
http://www.autoscout24.de/List.aspx
```

with the following search part

```
vis=1&make=9&model=16581&...
```

Now we first match the tags to the corresponding URL field and the string representation to the associated value, yielding

```
make=13
model=1664
```

These two key-value pairs can then be inserted in the original search part, which gives us the new search part: `vis=1&make=13&model=1664&...`

Depending if a POST or GET request is required, the variable bindings are either encoded in the body of the message or directly in the URL.

After the HTTP request is send to the server, we either directly get back the result page, or alternatively an intermediate page. In the latter case we automatically navigate to the result page based on the *Page-Keyword-Action paradigm*, which is presented in the next section.

3. Deep Web navigation. The navigation model is a crucial part of our system. Based on the model the system can determine anytime, if it has already reached the result page or if it is on an intermediate page. Additionally the model determines the actions, which should be performed for a specific intermediate page, e.g. to click on a link or fill out a new form field. The key idea of our Page-Keyword-Action paradigm is that the system first determines its location (intermediate vs. result page) based on a page keyword and then invokes a series of associated actions if appropriate.

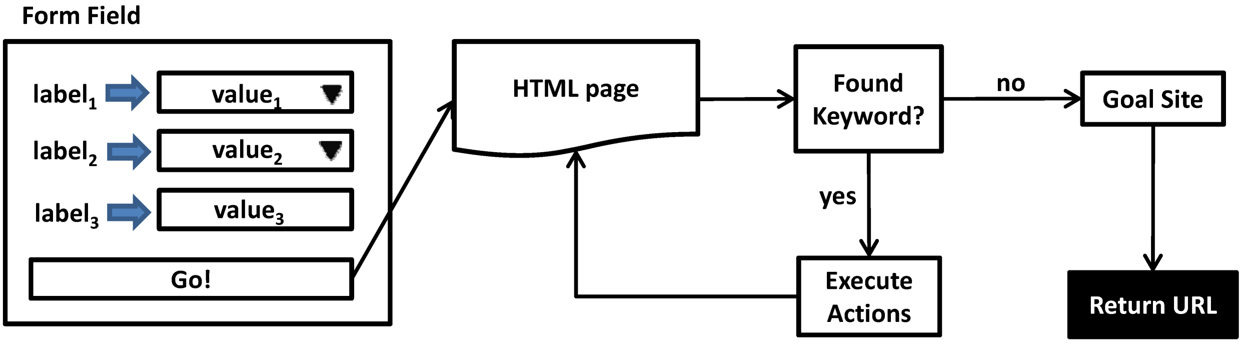
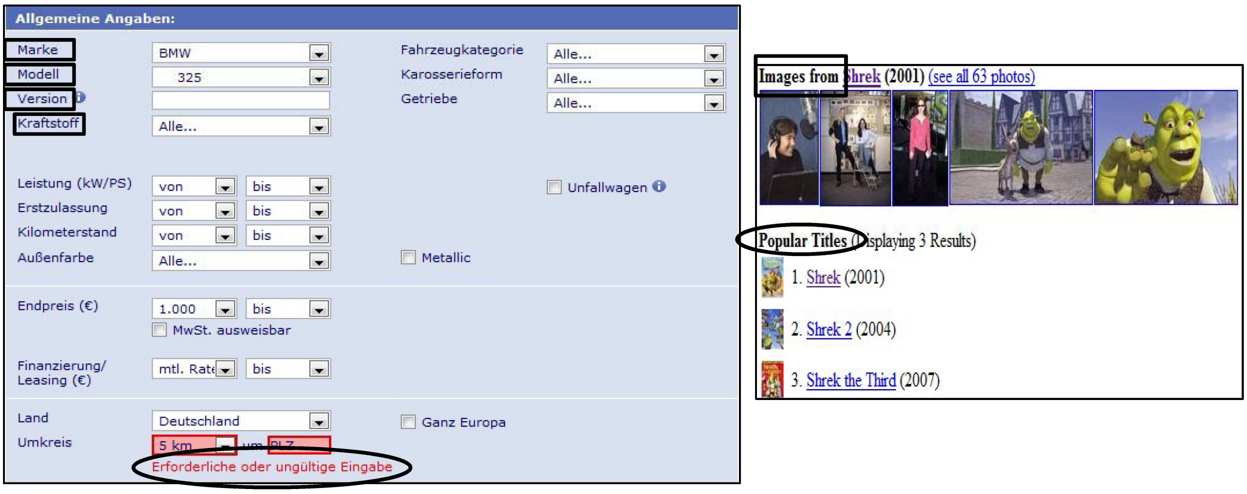


Fig. 3.1. *Navigation process*

3.1. Deep Web Navigation. The overall navigation process is illustrated in Figure 3.1: the user provides the system with a value map that contains for each desired input element label/value combinations. If the form field contains dynamic input elements for which she has provided input label/value combinations we check if they are legal. If so, we subsequently fill out and submit the form field with these combinations, which yields a new Web page (additionally, we use the information obtained during form analysis for directly generating the request POST/GET URL; thereby we can offline mimic the behavior of the form field). For this Web page we check, if we can find one of our defined keywords (cf. Section 3.2). If so, we perform the associated actions which result in a new Web page and check again if we are on a intermediate page. The cycle continues as long as we can find keywords on the Web page. To avoid an infinite loop, the user can specify an upper bound on the number of possible intermediate pages, after which an error message is returned. If we cannot find a keyword on the current Web page, we have found the goal page and return its URL.

3.2. Intermediate Page Keyword. Deep Web pages are typically created dynamically, i. e. data from a background database is filled into a predefined presentation template. Therefore, we can usually identify fixed elements, which are part of the template and are almost identical between different manifestations. After the form analysis is finished the user can iteratively submit the form with different options. If an input value combination leads her to an intermediate page, she can identify the relevant keyword as described in the following. If she has already reached a result page for a value combination no further user interactions are required. Note that as long as she is in the context of the currently active form field, she can also access a series of intermediate pages and for each page specify a series of actions. For the identification of a specific intermediate page we opted for a static text field. The reason is that it can be included in many HTML elements, e.g. the div, h2, or the span tag and given our template assumption they serve as a sufficient discriminatory factor. Other more advanced techniques based on visual markers on the page or more IR-related techniques, such as text classification approaches [19], could be used in this context as well and are planned as future work. In Figure 3.2 we have marked potential candidates for keywords with a rectangle.



Allgemeine Angaben:

Marke	BMW	Fahrzeugkategorie	Alle...
Modell	325	Karosserieform	Alle...
Version		Getriebe	Alle...
Kraftstoff	Alle...		
Leistung (kW/PS)	von bis	□ Unfallwagen	
Erstzulassung	von bis		
Kilometerstand	von bis		
Außenfarbe	Alle...	□ Metallic	
Endpreis (€)	1.000 bis □ MwSt. ausweisbar		
Finanzierung/ Leasing (€)	mtl. Rate bis		
Land	Deutschland	□ Ganz Europa	
Umkreis	5 km um PLZ Erforderliche oder ungültige Eingabe		

Images from Shrek (2001) (see all 63 photos)

Popular Titles (Displaying 3 Results)

1. Shrek (2001)
2. Shrek 2 (2004)
3. Shrek the Third (2007)

FIG. 3.2. *Intermediate pages for* `http://www.autoscout24.de` *(left) and* `http://www.imdb.com` *(right)*

The most likely candidates which are most characteristic are encircled with an ellipse, e.g. the error message for the car search service shown on the left. After the user has identified the keyword in the page, she can now specify actions that should be performed in order to reach the result page.

3.3. Intermediate Page Actions. The above specified keywords can be used to identify intermediate pages. However, our ultimate goal is to find a result page given a set of input value combinations for the initial form field. Therefore some actions, such as clicking on a link or filling out and submitting a new (intermediate) form, have to be performed to access the next—preferably result—page. In order to uniquely identify the appropriate HTML elements on which the stored actions should be executed, we defined a path addressing language called KApath, which is a semantic subset of XPath [5]. In order to access the appropriate action element, the system first finds the common ancestor of the keyword element and the action element and then descends downwards in the action element branch. Afterwards, the registered actions are executed for the found action element. Thus, KApath supports the following path expressions:

- **/Node[@aname$_1$=avalue$_1$=] . . . [@aname$_n$=avalue$_n$]**: The element in the DOM tree that matches the specified attribute name-value combinations of type Node,
- **/P**: Immediate parent node of current node,
- **/P::P**: All (transitive) parent nodes of current node,
- **/P::P/Node[@aname$_1$=avalue$_1$] . . . [@aname$_n$=avalue$_n$]**: The first found parent node in the DOM tree that matches the specified attribute name-value combinations starting from the current node and is of type Node,
- **/Child**: Immediate child nodes of current node,
- **/Child::Child**: All (transitive) child nodes of the current node,
- **/Child::Child/Node[@aname$_1$=avalue$_1$] . . . [@aname$_n$=avalue$_n$]**: The first found child node in the DOM tree that matches the specified attribute name-value combinations starting from the current node and is of type Node.

Figure 3.3 shows an example how the associated action element in a page can be referenced with respect to the page keyword with a KApath expression. Here, the TBODY node is the first common parent node for both (keyword and action) elements. Therefore the system automatically generates a KApath expression which allows optional intermediate elements between the keyword and the first common parent node. For finding the correct action element it is crucial to consider its attributes as well.

If the desired action elements have no (e.g. links) or dynamic attributes (e.g. visibility), we additionally store the absolute path from keyword to action element and the tree structure starting from the common parent. Another situation where we can make use of the absolute path is when the HTML page structure has changed and the common parent node is still on the same level in the DOM tree but in another branch. The tree structure is helpful if there are changes on the way downwards from the common parent node.

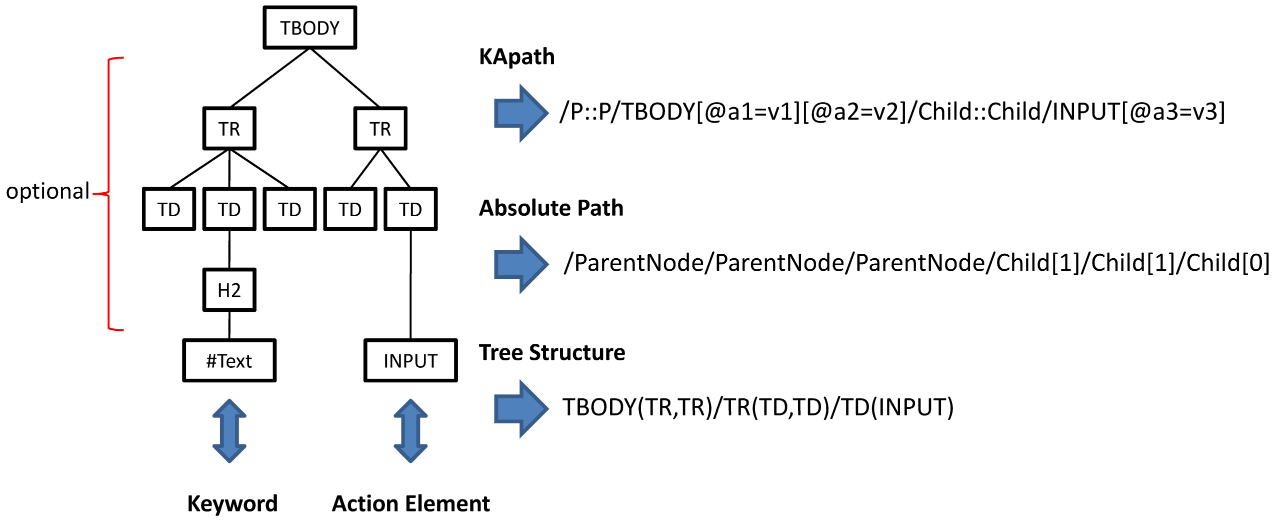

TBODY
TR
TR
optional
TD
TD
TD
TD
TD
H2
#Text
INPUT
Keyword
Action Element
KApath
/P::P/TBODY[@a1=v1][@a2=v2]/Child::Child/INPUT[@a3=v3]
Absolute Path
/ParentNode/ParentNode/ParentNode/Child[1]/Child[1]/Child[0]
Tree Structure
TBODY(TR,TR)/TR(TD,TD)/TD(INPUT)

Fig. 3.3. *Example KAPath expression, which allows optional HTML elements in the intermediate page*

3.4. Recording User Actions. Based on the user's browsing behavior, the system can generate the complete navigation model. First, she identifies the keyword for an intermediate page by clicking on the relevant text in the Web page. Then, the system determines the closest surrounding HTML element and stores the relevant context information. Afterwards, the system monitors the user behavior and stores each action she performs until she reaches a new page. Based on this action log, the system can automatically determine the paths and tree structures for each action.

To ease the recording of the user actions we have implemented *WScript*, a HTML action language similar to Chickenfoot [8]. This intermediate script language is convenient, because in order to find the HTML elements on which the actions have to be invoked we have to rely on the navigation structures defined in Section 3.3. Therefore, the provided actions have a navigation and (if applicable) an input part.

The following types of actions are supported by our system:

- Clicking on links: *link(absolute path, KApath, tree structure)*,
- Entering text in input fields: *enter(absolute path, KApath, tree structure, element name, element ID, input value)*,
- Selecting a checkbox or radio button: *click(absolute path, KApath, tree structure, element name, element ID)*,
- Selecting an option from a dropdown menu: *dropdown(absolute path, KApath, tree structure, option text, element name, element ID)*, and
- Submitting forms: *click(absolute path, KApath, tree structure, element name, element ID)*.

The element name and ID that are present for some actions are identical to the name and ID attributes of the underlying HTML element and are used first to find the relevant HTML element. If the lookup by ID and name fails, the search for the action element continues with the KApath as described in Sections 3.3 and 4.2. For example the following action expression would enter *Hallo World* into the text input field of the HTML tree in Figure 3.3:

```
enter(/ParentNode/ParentNode/ParentNode/Child[1]/Child[1]/Child[0]
,/P::P/TBODY[@a1=v1][@a2=v2]/Child::Child/INPUT[@a3=v3],
TBODY(TR,TR)/TR(TD,TD)/TD(INPUT),,,Hallo World)
```

Together, keyword and the associated actions form the navigation model for this intermediate page (cf. Figure 3.3).

4. Implementation. In this section, we describe in detail the implementation of DNavigator. Because the framework is geared towards casual Web users, important requirements must be met, most notably the tool must be easy to use. The DNavigator functionality is implemented as a Firefox extension in Java and JavaScript running a MySQL database for storing the necessary metadata (cf. Figure 4.1). LiveConnect [11] provides JavaScript with the ability to create and manipulate standard Java objects so that the system can connect to the database, e.g. to store the extracted dynamic dependencies and the navigation model, and fetch the predefined navigation models from the database to manipulate an intermediate page.

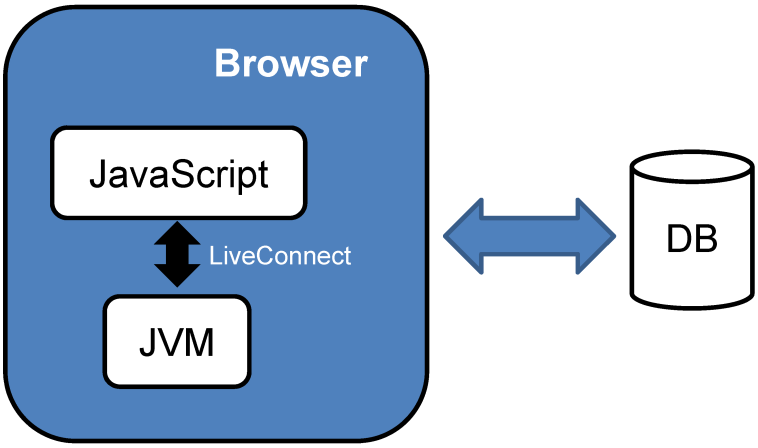


Fig. 4.1. *System architecture*

The rest of this section describes the implementation as well as the main issues we solved while implementing the system.

4.1. Navigation Model Creation. The system tracks a user's navigation actions on an intermediate page by adding JavaScript event handlers to Web pages before recording. These event handlers are invoked when certain user actions occur (e.g., clicking on text, clicking on a link, changing the selected option in a dropdown menu, etc.), which are supported by our system. The recording process is as follows: when the user presses the analysis button in the Firefox plugin window, the system sets event handlers on all clickable elements in the page displayed in the browser (i. e., handlers for links, handlers for forms, etc.). When an event fires, the system records all the necessary information for the event, e.g. KApath, absolute path, tree structure etc. It must then wait until the following page is loaded to repeat the process of adding handlers and waiting for events.

In order to determine the KApath, absolute path and tree structure with respect to the keyword and action element the system traverses the Document Object Model [1] tree starting from both elements.

4.2. Deep Web Navigation. After submitting the offline generated HTTP request, the first web page is returned from the target server. The system inserts an *onload* handler in the Web page to detect when the page has been completely loaded. Then, after the page has been downloaded, the navigation is invoked, i. e. the system will check whether one of our predefined keyword elements exist in this returned page. If this is the case, it is an intermediate page. Because for each keyword element we have saved its HTML type, attributes sequence and contained text, the check process was realized in JavaScript using the document object and node object based on the DOM tree, i. e. with the method *getElementsByTagName()*. The system first finds all HTML elements that have identical HTML types as keyword element. After the comparison between the attributes of these found elements and the stored attributes of the keyword element, and between the saved keyword text and the found keyword text, the system can determine whether a saved keyword (keyword element) exists in this intermediate page.

For any intermediate page a number of related actions must be performed, so that the system is able to navigate in the direction of the result page. Before such actions are executed, the system must first find the action-related elements, i. e. we must find all HTML elements, on which the action-events have to be activated. For this goal we use WScript that was presented in Section 3.4. The associated script will be fetched from the database using the Website ID and the identified keywords. Afterwards, an interpreter function is invoked to parse and execute every WScript expression step by step. Here, we iteratively use the following approaches:

1. When the corresponding element's attributes *id* or *name* are available, the action element can be easily found with the method *getElementById()* and *getElementsByName()*.

2. Otherwise, we try to find the action element based on the KApath-expression.
3. Finally, if the action element after executing the first two strategies cannot be found, the system uses the absolute path and the tree structure to locate the action element.

The execution of related actions is simulated using DOM Level 2 events [1, 2], i. e. fake event objects are created using the *document.createEvent()* method. Afterwards, they are activated on the desired action element using the *element.dispatchEvent()* method.

5. Evaluation. In our experiments, we evaluated the following aspects for our two major components: accuracy and runtime. For this, we selected 100 Deep Web sites from different domains, e.g. car search and video search. 60 of them were directly adopted from the website table in [7], because they contain a large amount of data. The others were selected by a focused search on Google on Deep Web repositories. For a full list of the tested Web sites we refer the interested reader to [25].

5.1. Experimental Results. All experiments have been conducted on a Thinkpad T60 (Intel Core Duo 2 Processor T7200 2,00Ghz with a 667MHz front side bus and 2GB of main memory) running Windows Vista, MySQL Server 5.0, Java JDK 1.6 and Firefox 2.0.0.12. The maximal download rate of the internet connection was 2048 Kbit/s and the maximum upload rate 256 Kbit/s.

5.1.1. Frontend Analysis. For 99% of the tested Web sites the frontend analysis was successful, finding the correct static and dynamic dependencies. Depending on the number of items in the dropdown menus of the form fields, the time needed for analysis took from 0.5 to 30 seconds, i. e. 4.28 seconds on average. Since this analysis has only to be performed once, we feel that performance optimizations for this analysis are of limited benefit, because our major focus is on correctly identifying hidden dependencies between the dropdown menus.

Table 5.1
Time (in seconds) for navigation experiments.

# Int. Pages	# Web Sites	Page Load	1 Model	6 Models
0	58	2.25	2.26	2.31
1	22	-	4.60	4.66
2	14	-	6.47	6.55
3	4	-	8.12	8.23
4	1	-	9.70	9.83
5	1	-	11.06	11.22

5.1.2. Deep Web Navigation. For 96% of the tested Web sites we were able to successfully find a keyword and to navigate to the desired result page. The navigation process took from 2.26 to 11.22 seconds, i. e. 3.79 seconds on average. As shown in Table 5.1 most of the time was spend for loading pages, i. e. 2.25 seconds on average. The columns labeled *1 Model* and *6 Models* indicate the number of registered navigation models for each page. As can be seen, the overhead for checking multiple models was marginal in contrast to the time spent for loading pages. This is due to the fact that the execution of the actions is performed by the browser on the client side and since no computationally intensive algorithm is required to identify intermediate pages.

5.2. Open Issues. Our evaluation revealed the following open issues of our system.

5.2.1. Frontend Analysis.

- Delayed AJAX interactions: For one Web site we were unable to correctly detect the dynamic dependencies because the server took longer than our specified threshold to change the items in the respective dropdown menu.

This could be remedied by increasing our threshold value to some extent, but further investigation is needed to find a general solution for this problem.

5.2.2. Deep Web Navigation.

- Dynamic request URLs: Usually, different request URLs only differ in the searchpart, i.e. the part of the URL after ?, due to different variable bindings, which are transferred to the server. Two Web sites in our test bed used different paths as well, which our system converts into illegal request URLs.

- Hidden form elements: Since the user can only drag labels to visible form elements, values in hidden form elements that have to be correlated with visible elements cannot be detected by our system.
- Session IDs: Session IDs are often used to track user interactions with Web pages and are only valid for a certain period. Because we are not able to produce a new (fake) session ID for each service, the offline generated URL becomes invalid over time.

All of the abovementioned issues could be solved by filling out the frontend form at runtime and skipping the offline generation of the URL for such resources.

- Static URLs: Our system determines, if a new Web page has been loaded based on the current URL. If the URL does not change after a form has been submitted, we are not able to initiate the navigation process and add the required event handler as described in Section 4.2.

This can be solved by using another metric for determining if a new Web page has been loaded, e.g. a checksum of the Web page.

6. Related Work. [22] presents a framework called DEQUE for querying Web forms where input values are allowed from relations as well as from result pages. As a part of their system they also model Web form interfaces, but their focus is more on the modeling of consecutive forms and they did not consider the dependencies between form input elements.

A number of navigation concepts have been proposed for accessing Deep Web sources. [10] and [18] proposed process-oriented navigation maps, which describe a set of paths from a start page to a result page. But these maps rely on consecutive state transitions and fixed interactions between them. In [16] the user actions from a specified start page over possibly multiple intermediate pages to an end page are recorded in a navigation map. The actions that link two adjacent pages are strongly connected as well. A sophisticated Deep Web navigation strategy based on the branched navigation model is proposed in [6]. The navigation is represented as a sequence of pages, with envisioned future support for standard process-flow languages such as WS-BPEL [4]. In [21] a navigation sequence was specified in NESQL [20]. The NESQL expression contains metadata about action elements, for instance, their specified names and types. Each expression will be interpreted based on these element properties. By storing historical information from previous accesses of a Deep Web resource and utilizing browser pools, their system tries to reuse the current state of a browser. [24] describe a system called WebVCR, which is able to record and replay a series of browser steps as a smart bookmark, but they do not consider optional intermediate pages.

Our framework is not dependent on a rigid sequence of intermediate pages, because for each new page all keyword patterns are checked and therefore the previous state of the system is not important for our page-oriented navigation model. Besides, we do not need a complex navigation algebra or calculus for the navigation process because we just save the above described navigation model for each intermediate page. For instance, the framework proposed by [10] relies on a subset of serial-Horn Transaction F-Logic [17]. As discussed in Section 3.4, the saved action sequences are just macro procedures, which are interpreted by our JavaScript macro engine.

7. Future Work. At the moment we only perform a hard string match between user inputs and the options in a dropdown menu. If the strings do not match exactly an error is returned. At the moment we are investigating approximate string matching techniques [9] to alleviate this problem to some extent. An alternative would be to use semantic similarity metrics, such as proposed in [27], which would also be able to capture the similarity between the two car companies *Toyota* and *Lexus* (a division of Toyota). The work by [26] tries to automate the extraction of query capabilities, such as labeling form input elements and finding legal ranges of input values. This could be interesting to combine with our approach to suggest tags to the user, or to try to match the labels on the Web form with the tags in the user vocabulary and thus easing the labeling of the Web forms.

Our experiments suggest that the determination of a suitable keyword is crucial for the successful identification of an intermediate page, and that for some cases it might be better to skip the offline generation of the start URL. Currently, we are extending our research prototype to accept a list of keywords and work on an algorithm to automatically suggest meaningful and discriminatory keywords. Ultimately, we are interested in generalizing the concept of immediate page identification to more elaborate techniques, such as the visual appearance of the Web page.

8. Conclusion. In this paper we presented DNavigator, a framework for accessing result pages of Deep Web sites, which contributions are twofold: first, a frontend analysis has been described, which needs only to be performed once, and afterwards the system can simulate the behavior of the Web form offline. Second, we have proposed a simple but effective Deep Web navigation strategy, which replaces a heavy-weight navigation calculus with an intermediate page identification procedure and a set of actions that navigate to the next page. The proposed navigation strategy has the following benefits:

1. *It is stateless.* Because for each page, we check all available navigation models, we are not dependent on a specific navigation order.
2. *Simple extensibility.* If the system encounters a new and so far unknown immediate page, the user can easily extend the existing navigation model with only a few steps.
3. *Simple presentation of the model.* Each navigation model has an intuitive textual representation which is easier to understand and use than a complicated navigation calculus.

To sum up, DNavigator offers a simple user interface, but successfully deals with most of the problems that are posed by real-world Deep Web sites as our evaluation has shown.

REFERENCES

[1] *Document object model (dom).* `http://www.w3.org/DOM/`

[2] *Document object model (dom) level 2 events specification.* `http://www.w3.org/TR/DOM-Level-2-Events/`

[3] *Hypertext transfer protocol—http/1.1 (rfc 2616).* `http://tools.ietf.org/html/rfc2616/`

[4] *Web services business process execution language version 2.0.* `http://www-128.ibm.com/developerworks/library/specification/ws-bpel/`

[5] *Xml path language (xpath) version 1.0.* `http://www.w3.org/TR/xpath`

[6] R. Baumgartner, M. Ceresna, and G. Ledermüller, *Deep web navigation in web data extraction*, in Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce., Vienna, Austria, 2005, pp. 698–703.

[7] M. K. Bergman, *The deep web: Surfacing hidden value, white paper.* `http://www.brightplanet.com/images/stories/pdf/deepwebwhitepaper.pdf` 2001.

[8] M. Bolin, M. Webber, P. Rha, T. Wilson, and R. C. Miller, *Automation and customization of rendered web pages*, in Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, Seattle, WA, USA, 2005, pp. 163–172.

[9] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, *A comparison of string distance metrics for name-matching tasks*, in Proceedings of the Workshop on Information Integration on the Web, Acapulco, Mexico, pp. 73–78.

[10] H. Davulcu, J. Freire, M. Kifer, and I. V. Ramakrishnan, *A layered architecture for querying dynamic web content*, in Proceedings of the ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, USA, 19999, pp. 491–502.

[11] D. Flanagan, *JavaScript: The Definitive Guide, Fourth Edition*, OďReilly, Sebastopol, CA, USA, 2001.

[12] B. He, M. Patel, Z. Zhang, and K. C. C. Chang, *Accessing the deep web*, Communications of the ACM, 50 (2007), pp. 94–101.

[13] H. He, W. Meng, C. T. Yu, and Z. Wu, *Wise-integrator: A system for extracting and integrating complex web search interfaces of the deep web*, in Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 2005, pp. 1314–1317.

[14] P. Heymann, G. Koutrika, and H. Garcia-Molina, *Can social bookmarking improve web search?*, in Proceedings of the International Conference on Web Search and Web Data Mining, Palo Alto, California, USA, 2008, pp. 195–206.

[15] T. Hornung, K. Simon, and G. Lausen, *Mashing up the deep web—research in progress*, in Proceedings of the 4th International Conference on Web Information Systems and Technologies, Funchal, Madeira—Portugal, 2008, pp. 58–66.

[16] N. Julasana, A. Khandelwal, A. Lolage, P. Singh, P. Vasudevan, H. Davulcu, and I. V. Ramakrishnan, *Winagent: A system for creating and executing personal information assistants using a web browser*, in Proceedings of the 2004 International Conference on Intelligent User Interfaces, Funchal, Madeira, Portugal, 2004, pp. 356–357.

[17] M. Kifer, *Deductive and object-oriented data languages: A quest for integration*, in Proceedings of the 4th International Conference on Deductive and Object-Oriented Databases, Singapore, 1995, pp. 187–212.

[18] J. P. Lage, A. S. da Silva, P. B. Golgher, and A. H. F. Laender, *Collecting hidden web pages for data extraction*, in Proceedings of the 4th ACM CIKM International Workshop on Web Information and Data Management, Virginia, USA, 2002, pp. 69–75.

[19] K. Nigam, A. McCallum, S. Thrun, and T. M. Mitchell, *Learning to classify text from labeled and unlabeled documents*, in Proceedings of the 15th National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, Wisconsin, USA, 1998, pp. 792–799.

[20] A. Pan, J. Raposo, M. Alvarez, J. Hidalgo, and A. Vinaet, *Semi-automatic wrapper generation for commercial web sources*, in Proceedings of the Working Conference on Engineering information Systems in the Internet Context, Kanazawa, Japan, 2002, pp. 265–283.

[21] J. Raposo, M. Alvarez, J. Losada, and A. Pan, *Maintaining web navigation flows for wrappers*, in Proceedings of the 2nd International Workshop on Data Engineering Issues in E-Commerce and Services, San Francisco, CA, USA, 2006, pp. 100–114.

[22] D. Shestakov, S. S. Bhowmick, and E. P. Lim, *Deque: Querying the deep web*, Data & Knowledge Engineering, 52 (2005), pp. 273–311.

[23] K. Simon and G. Lausen, *Viper: Augmenting automatic information extraction with visual perception*, in Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, 2005, pp. 381–388.

[24] A. Vinod, J. Freire, B. Kumar, and D. F. Lieuwenet, *Automating web navigation with the webvcr*, in Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, 2000, pp. 503–517.

[25] Y. Wang, *Deep web navigation by example*, master's thesis, Institute of Computer Science, Albert-Ludwigs University Freiburg, 2008.

[26] Z. Zhang, B. He, and K. C. C. Chang, *Understanding web query interfaces: Best-effort parsing with hidden syntax*, in Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, 2004, pp. 107–118.

[27] C. N. Ziegler, K. Simon, and G. Lausen, *Automatic computation of semantic proximity using taxonomic knowledge*, in Proceedings of the ACM CIKM International Conference on Information and Knowledge Management, Arlington, Virginia, USA, 2006, pp. 465–474.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: January 11th, 2008
Accepted: March 19th, 2008
Extended version received: June 25th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 293–301. http://www.scpe.org

ISSN 1895-1767
© 2008 SCPE

DERIVING A LIGHTWEIGHT CORPORATE ONTOLOGY FORM A FOLKSONOMY: A METHODOLOGY AND ITS POSSIBLE APPLICATIONS

CÉLINE VAN DAMME[*], TANGUY COENEN[†] AND EDDY VANDIJCK[*]

Abstract. Companies use company-specific terminology that may differ from the terminology used in existing corporate ontologies (e.g. Tove) and therefore need their own ontology. However, the current ontology engineering techniques are time-consuming and there exists a conceptual mismatch among developers and users. In contrast, folksonomies or the flat bottom-up taxonomies constituted by web users' tags are rapidly created. In this paper, (1) we present an approach that cost-efficiently derives a lightweight corporate ontology from a corporate folksonomy, (2) by means of a folksonomy dataset from a European company, we provide preliminary evidence that our suggested approach reflects the company-specific terminology, (3) we detect a number of possible applications for the company when implementing the presented methodology on a corporate folksonomy and (4) as an additional evaluation, we asked the company to briefly evaluate the results and possible applications.

Key words: ontology, folksonomy, company, applications

1. Introduction.

It has been stated, e.g. in [24, 6] that ontologies improve the communication among humans or machines since they provide a shared understanding of a domain. This makes that ontologies are very useful for companies. For instance they can help to improve the communication between employees.

At this moment, there exist several corporate ontologies, for instance Tove [7] and Enterprise ontology [26]. These ontologies describe general concepts and relations related to enterprise and process modeling. We believe these kinds of ontologies may not be useful for every enterprise since companies have a corporate-specific terminology and consequently have their own concepts. In our opinion, an enterprise may need its own corporate ontology.

Building ontologies with the current ontology engineering techniques have disadvantages. First of all, it is a very time-consuming process [2] and secondly the actual users are not involved in the developing process. As a consequence there exists a conceptual mismatch between the developers and the actual users' vocabulary [11].

These disadvantages are not present in the relatively new categorization method called tagging and its resulting folksonomy. Following the Web2.0 paradigm, a growing number of websites incorporate a tagging/folksonomy mechanism. They allow users to refer to resources (bookmarks, pictures or scholarly publications) on the web with freely selected keywords or tags. The users are not restricted to a controlled vocabulary produced by a group of experts. Users can enter any words that enter their mind. This makes them active participators in creating new tags. Aggregating this user created meta data leads to a flat, bottom-up taxonomy, also known as a folksonomy.

Despite the strengths, tagging has its weaknesses: no conceptual meaning or hierarchical relations are added to the tags. As a consequence, tags have no synonyms or homonyms. Furthermore, specialized as well as general tags can be used to annotate the same resource [9, 10]. These weaknesses can be solved by (1)giving the users tools that enable them to add more information to their tags (e.g. cluster tags as on Delicious) [10] and/or (2) trying to generate more information on the tags by employing text mining, statistical techniques and asking additional feedback from the community [4].

The last few years, we observe a growing attention of the semantic web community for tagging and its resulting folksonomies. At the one hand, we observe researchers that try to enrich the flat ambiguous tags with existing online resources (e.g. Google, Wordnet, existing ontologies) [22] and on the other hand, there are researchers that consider this user created meta data as a valuable source to develop ontologies [4].

In this paper, we argue that cost-efficiently deriving a lightweight ontology from a folksonomy is also applicable to a corporate folksonomy. We regard a lightweight ontology as the simplest form of an ontology: an ontology where only one relation is included or a taxonomy as described by [25]. We propose a 6-step approach which includes several techniques such as the Levenshtein metric, co-occurrence, conditional probability, transitive reduction and visualization. Although, some suggestions have already been made on how a corporate ontology can be built from a corporate folksonomy [3], no research results have been published so far. We implemented our approach on a corporate folksonomy of a large European distribution company in which Dutch and French are the two official company languages. We obtained the simplest form of an ontology, a lightweight

[*] MOSI, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium, {cvdamme, eddy.vandijck}@vub.ac.be
[†] STARLab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium, Tanguy.Coenen@vub.ac.be

ontology, and visualized it with the open source tool Graphviz (`http://www.graphviz.org/`). By means of the generated lightweight ontology, we were able to detect other possible applications than the one to improve the communication among the employees in the company. As an additional evaluation, we asked the company to evaluate the results and its applications.

The paper is structured as follows: we provide an overview of related work in section 2. In section 3, we discuss all the techniques of the methodology and explain how they can be integrated in our 6-step approach. In section 4, we elaborate on the corporate folksonomy dataset discuss the general results of applying our approach to the dataset. We describe possible applications of the approach for the company in section 5. Section 6 discusses our findings and presents future research. A conclusion is provided in section 7.

2. Related Work. At the time of writing, few papers have been written on discussing the use of folksonomies in a company. The authors in [17] present a social bookmarking tool, called Dogear, that lets employees tag their bookmarks from the corporate intranet and the World Wide Web. The advantages of collaborative tagging in the enterprise are discussed in [12]. The authors suggest that tagging can be used as an expert location tool that facilitates the process of organizing meetings with experts in the company. Tags are a reflection of people's interest and/or knowledge and can as a consequence be seen as a tool to detect experts and their domain of expertise.

However, the authors in [17, 12] do not explain how to make the tags less ambiguous nor turning them into an ontology. This is discussed in [3]. The authors propose to derive a CRM or Customer Relationship Management ontology from a corporate folksonomy. They suggest an integrated visual approach that integrates text mining techniques, tags and user feedback. Each time the employee adds a message or note to the CRM system, tags are required. At the same time, automatic keywords are detected based on the tf-idf score. The tf-idf score is calculated by multiplying the word's document frequency by the logarithm of its inverse document frequency in the set of relevant company documents. The higher the score, the more descriptive the keywords are [20]. In a first phase the user has to indicate whether there exists a relationship between the tags and the keywords with the highest tf-idf score. The relationship has to be specified in a second phase. In this approach, the human effort as well as the implementation time is very high. We also have to point out that the proposed approach still has not been tested.

Literature on folksonomies enrichment or turning folksonomies into ontologies is currently more common in the domain of the World Wide Web. In [21] tags of the photo-sharing site Flickr (`http://www.Flickr.com/`) were used in an experiment to induce a taxonomy, the simplest form of an ontology [25]. The approach of [21] is based on statistical natural language processing techniques where a subsumption or hierarchical relation was deducted. The authors of [22, 4] both suggest to include different techniques as well as the wealth of existing online web resources such as Wordnet, Wikipedia, Google, online dictionaries and existing ontologies. The authors in [22] present an approach to enrich tags with semantics to make it possible to integrate folksonomies and the semantic web. The authors use online lexical resources (e.g. Wordnet, Wikipedia, Google) and ontologies to map tags into concepts, properties or instances and determine the relations between mapped tags. However, the resources are tapped in one way (e.g. Wikipedia is used as spelling checker for tags) and the community is not involved to confirm the semantics obtained from existing ontologies and resources. Consequently, tags that reflect new concepts, relations or instances or new relations between tags are neglected. On the contrary, the opposite is suggested in [4]: ontologies are derived from folksonomies. Online lexical resources are suggested to be exploited in several ways. For instance Wikipedia is suggested as a spelling checker as well as a tool for finding concepts and homonyms. Furthermore, the authors suggest involving the community.

However, a corporate folksonomy differs from a folksonomy created on the World Wide Web. The users, their underlying motivations and the environment can be different. In case of a corporate folksonomy the user or employee is known and will not always tag voluntarily. An employee may be enforced to tag or may be given an incentive by the company. As a consequence, the amount of additional feedback asked from the users to create a lightweight ontology should be reduced. Labor costs are very high and therefore the number of employees involved with the feedback process should be minimized. In contrast to web communities it is far easier to ask the cooperation of the community: community members have a different mindset than employees and are more willing to participate in additional processes. However, in most cases they are anonymous. Company-specific terminology is mostly used in a closed company environment which makes it hard to include web resources in the ontology construction process. The terminology may contain terms which have a specific meaning for only a small group of employees.

3. Methodology. In this section, we first describe the different techniques we implement in the 6-step methodology, motivate why we do not include other techniques or online resources yet, and then elaborate on how we integrate the selected techniques as a whole.

3.1. Overview of techniques.

3.1.1. Levenshtein metric. The Levenshtein metric is a text similarity metric which calculates the distance between two words. More specifically, it counts how many letters have to be replaced, deleted or inserted to transform one word into the other [13]. It is a valuable technique to verify the similarities of two tags. In order to calculate the distance, first all possible tag pairs have to be made. In [22] a threshold value of 0.83 is used to indicate that two tags are similar. Yet tests showed us that a threshold value of 0.83 excluded a number of similar tags. For instance, the Dutch nouns *fiets* and *fietsen* or *bicycle* and *bicycles* in English, express the same thing but do not agree in number. Both tags are the same and their Levenshtein similarity is lower than 0.83. We believe this technique should be employed at a lower threshold value, we suggest 0.65, and include human feedback. A representative employee that is very well aware of all the terminology used in the company can be asked to confirm or reject the similarity.

As a tag cleaning method, we prefer this one to the one often suggested in literature, stemming. A stemming algorithm reduces tags to their stems or roots. The algorithm removes suffixes and hereby e.g. reduces the words *linked* and *links* to *link* [19]. The algorithm includes rules that are language dependent. Company-specific language can be lost because of the stemming algorithm. These words can differ from the general spelling rules or they can be abbreviations. Some languages, such as Dutch, incorporate English words in the vocabulary without adjustments to the Dutch language.

When stemming algorithms are used, there should be a way to determine the language of the tags and whether it involves corporate-specific language.

3.1.2. Co-occurrence. Luhn [14] stated that the frequency of words in a text can be used as a technique to detect relevant keywords for a document. Later, researchers in the domain of computational linguistics have started to use the statistical technique co-occurrence, the occurrence of two words used together in a text, to cluster terms [18]. [15] used a methodology based on co-occurrence to select the keywords for a document without a corpus or set of related documents. The co-occurrence technique is also proposed in the literature on folksonomies [21, 22]. For each tagged resource all the tag pairs are determined. The tie strength between a tag pair is increased each time two tags are used together.

It is interesting to know which tags are often used together to have already an idea which terms are often used together.

3.1.3. Conditional Probability. A rule based on the conditional probability definition was proposed in [16, 21]. More specifically, the rule tries to find out whether one of the tags in the pair can be defined as broader and the other one as narrower term. By applying the definition of the conditional frequency, the conditional probability is calculated by dividing the co-occurrence of the tag pair by the frequency of the individual tags. Results vary between 0 and 1. The higher the result, the more the term is used in combination with the other term and consequently the more depended it is of the other term. When the difference between the two results exceeds a certain threshold value, in [21] the threshold value is set to 0.8, a subsumption relationship is found.

Finding an appropriate threshold value should be determined based on trial and error testing.

3.1.4. Transitive Reduction. In [21] the authors remove the roots that are logically above the parent nodes. However, we believe transitive reduction, a technique from graph theory, is far more interesting. Transitive reduction reduces the edges of a graph G to a graph G' by keeping all the paths that exist between the nodes in Graph G [1]. The edges are consequently removed because of the implied transitivity.

3.1.5. Visualization Techniques. The use of visualization is proposed in [3] to lower the barriers to participate in naming the relations between concepts. In literature, several approaches for visualizing tags and lightweight ontologies are described. In [27] CropCircles are suggested to help people understand the complexity of a class hierarchy. We hypothesize that visualizing the lightweight corporate ontology may facilitate the validation process of the approach.

3.2. Other Techniques and online resources. Of course, a lot of other techniques (e.g. clustering techniques) or online resources could be interesting to extend the ontology with more relationships.

In [22, 4] the use of online resources such as Google, Wikipedia, online dictionaries is suggested as additional mean. The resources are regarded as spelling checkers and as a mean for retrieving concepts. The company-specific terminology makes it hard to use some of the sources on the internet. For instance, a company had a *gara* tag, used as the abbreviation of the Dutch word *garage*. When using *gara* as a search term for Google, we did not find any link referring to the correct meaning of the term. On Wikipedia, we found a page describing the term, but the concept or description attributed to it was incorrect. On Wikipedia, *gara* is a Basque word and the name of a Spanish newspaper. This causes problems. We have to know whether the tag belongs to the specific terminology of the company or not. In order to find this out, human feedback is necessary. However, asking employees to verify the word's background can quickly become too time-consuming. Therefore, we decided not to include any web resources yet.

3.3. 6-Steps Approach . Based on the techniques discussed in previous section, we explain how they can be integrated into our 6-step approach to derive a corporate ontology form a corporate folksonomy.

3.3.1. Step 1: Selection of the Tags. First, we remove all the Dutch stop words (Based on the list available at `http://snowball.tartarus.org/algorithms/dutch/stop.txt`) and filter the messages with fewer than 2 tags. We then withdraw the less frequently used tags by ranking the tags in an absolute frequency. Although in the domain of automatic indexing upper as well as lower bounds are used to exclude non-significant words, we assume that removing the upper bound tags will remove important company-specific elements for our lightweight ontology [8].

3.3.2. Step 2: Clean the Tags. Since folksonomies do not restrict its user to use a controlled vocabulary or predefined keywords, tags are polluted (e.g. plural and singular tags) and need to be cleaned up. We use the Levenhstein similarity metric combined with human feedback.

Based on a trial and error method, we decide to take 0.65 as a threshold value. All the tag pairs that reach a Levenhstein similarity of 0.65 will be presented and when two keywords are similar, the user has to check the corresponding check button, as visualized in figure one.

Then, the tag with the lowest frequency will be replaced with the one with the highest frequency. We opt for this rule since we believe that the tag with the highest frequency determines how the word should be written by the wisdom of the crowds in the company [23].

In figure 3.1, there are 4 tag pairs checked as similar. The tags with the highest frequency are always on the left. In the case of the tag pair (*winkel winkels*) or (*shop shops*) translated into English, the tag *winkels* will be replaced with *winkel* in the database. Whereas the tag pair (*artikel1234 artikel1235*) will not be adjusted. Latter tag pair contains dissimilar tags because they express different article numbers.

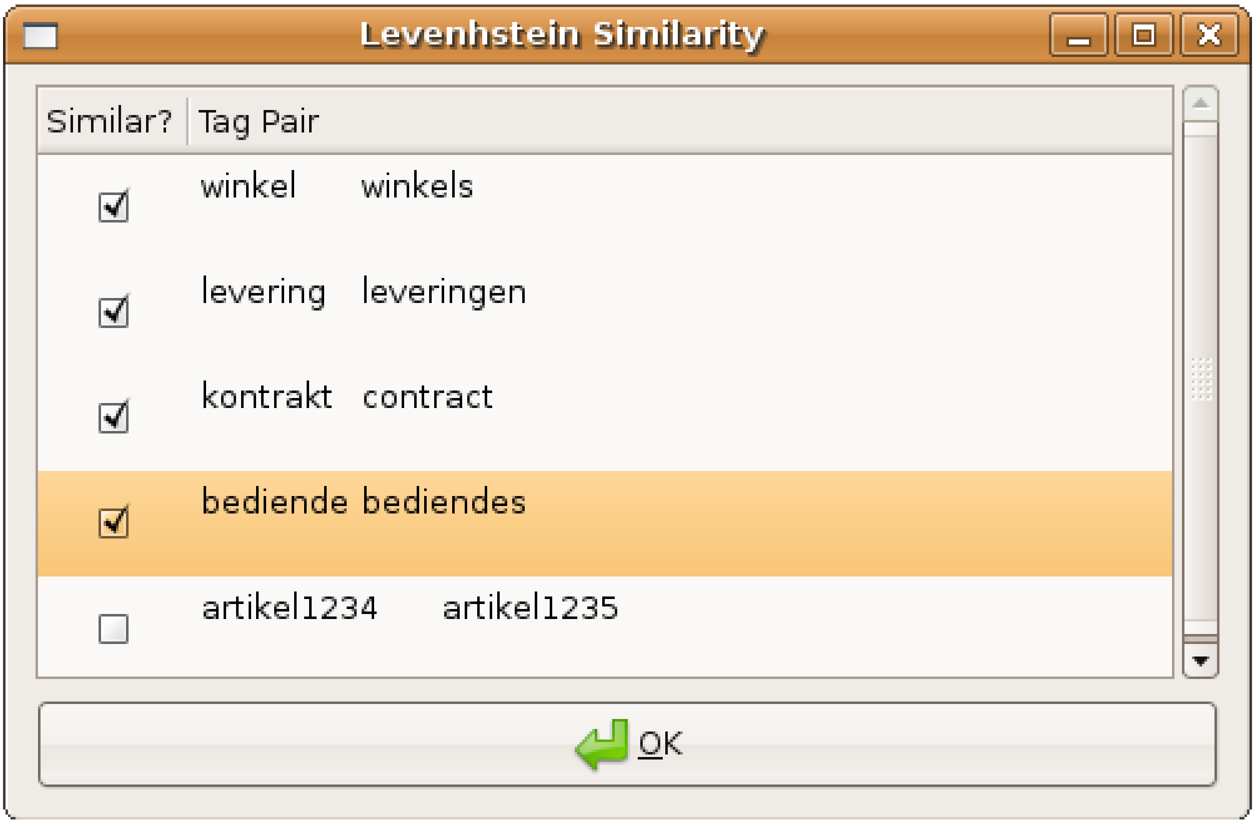
After the adjustment, we reselect the tags following the same procedure as described in the first step.

3.3.3. Step 3: Co-occurrence. For each message we make all the tag pairs. Then, we count the frequency of each unique tag pair. The more two tags are used together, the higher this frequency or co-occurrence value. Again, we decide to include only the ones with the highest frequency to find the most frequent relations.

3.3.4. Step 4: Finding Broader/Narrower Relations. We want to derive the simplest form of an ontology and therefore need to find the broader/narrower relations between the terms, for instance the relation between *animal* and *dog*. We apply the conditional probability function as described in previous section. Therefore, we divide the co-occurrence of the tag pair by the frequency of the tag itself. We did some manual tests deciding on 0.70 as the most appropriate threshold value. The higher the threshold value, the broader and the less deep the resulting ontology will be. For instance, when the tag pair *animal dog* occurs a 100 times and the frequency of both tags is respectively 500 and 120, we obtain the following results: animal = 0.2 and dog = 0.83. The tag dog exceeds the threshold value of 0.70 and therefore the relation between *animal-dog* can be considered as a broader narrower relationship.

3.3.5. Step 5 & 6: Transitive Reduction and Visualization. First, we apply the transitive reduction and then we visualize the remaining relations through Graphviz.

4. Dataset. In this section, we present the corporate folksonomy dataset and explain the results of applying our approach to this dataset.



Similar?	Tag Pair	
☑	winkel	winkels
☑	levering	leveringen
☑	kontrakt	contract
☑	bediende	bediendes
□	artikel1234	artikel1235

Fig. 3.1. *Asking human feedback based on the Levenshtein metric*

4.1. Description corporate folksonomy. We have implemented our approach in a large European distribution company with headquarters in Belgium in which Dutch and French are the two official company languages. The company employs more than 15.000 people across Europe.

Tagging has been used on all their communication messages for more than 20 years. Messages such as letters and faxes that are not sent electronically are manually scanned, tagged and archived into an information system. Tags replace the subject line of the message. Tagging is completely integrated in the corporate culture. The messages can be created manually, automatically and semi-automatically. The automatic and semi-automatic messages have default tags. In case of semi-automatic messages, the author has to add complementary tags. Manually created messages require user created tags.

Initially, tags were introduced to solve the information retrieval problem since full text search engines were not available at the time. Tagging has remained part of the communication messaging system. However, the ambiguity of the flat tags and the information overload obstructs the search process. The company introduced some tag rules such as a minimum number of tags, no stop words, no plurals and no conjugated verbs, but only a minority of the employees in the company obeys all these rules.

Even though the tagging system at this company is somewhat different from current web-based tagging practices, the 20-years worth of tagged messages represented a real opportunity to test out the approach in a real-life case. Such cases are rare, as not many organizations have adopted tagging in a way which allows the analysis of a large body of tags. Tagging is so widely adopted and part of the corporate culture we believe the tags can be made to represent a non-toy lightweight ontology.

4.2. Tag datasets. In 2006, more than 8.000.000 messages were created and roughly 60.000.000 tags in total were used. 91% of the messages are created by Dutch speaking employees.

Due to the large size of the dataset and limited computer power, we decided to make a selection of the tags. We focused our analysis on the tags added to Dutch messages. More specifically, we analyzed 2 different message types individually: quick internal messages and notes since these are often used message types in the company.

As we discuss in the following paragraphs, we split the dataset into two sets and applied the 6-steps approach to tags annotated to quick internal and notes message types from both datasets.

4.2.1. Tag dataset 1: tags from automatic, semi-automatic and manual messages. At the beginning, we were not able to make a distinction between tags from automatic, semi-automatic and manual messages since a unique field to filter out the manual ones is not stored by the company. Therefore, the first tag dataset consisted of tags from the automatic, semi-automatic and manual messages.

Some information systems in the company can send automatic messages to the employees to inform them on certain issues, for instance an employee confirms to be present at a certain meeting and the system automatically sends a message to the person who organized the meeting. Tags are automatically generated and added to the message. In the case of semi-automatic messages, a message is based on an existing template including a list of tags that have to be extended. Whereas in the case of manual messages, the message as well as the tags are manually created.

We applied the approach to this dataset and after tag cleansing, we selected a group of tags (approximately 150) with a very high frequency (between 5000 and 147.000) to grasp the meaning and interrelations of these frequently used tags. We did the same for the selection of tag pairs.

In figure 4.1, a part of the obtained lightweight ontology of the quick internal messages is visualized. We renamed the top level node "name_of_shop" to guarantee the anonymity of the company.

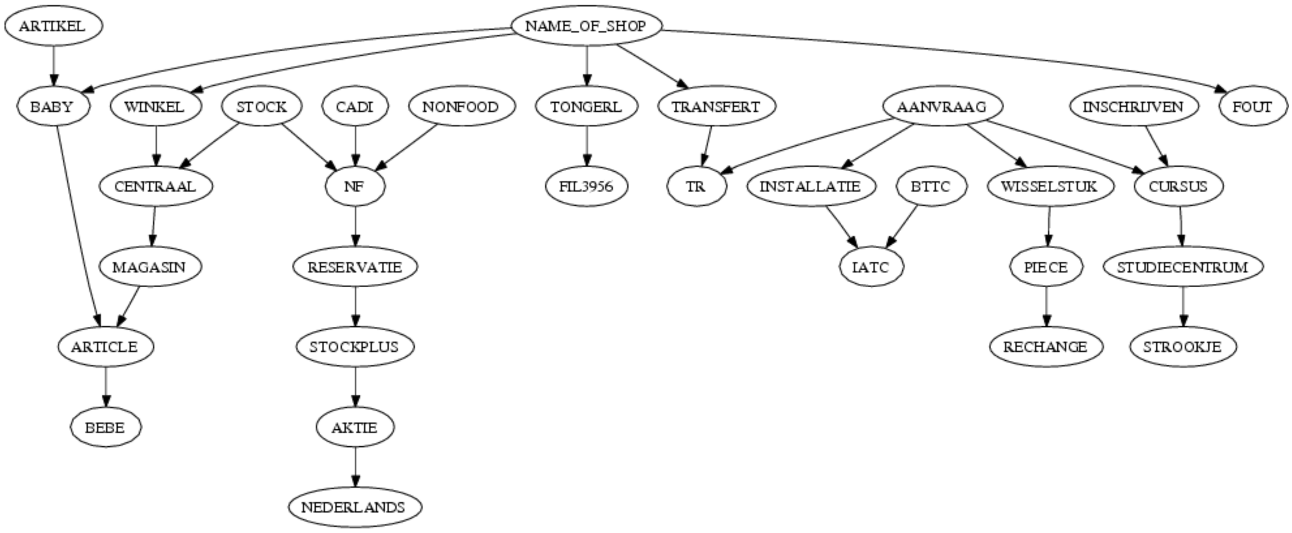


FIG. 4.1. *Partial results obtained from analyzing the quick internal messages from dataset 1*

4.2.2. Tag dataset 2: tags from manual messages. After presenting and discussing former results at the company, we realized it would be interesting to filter out the manual created tags. Apparently, many messages are automatically created and therefore partially influence the results received through previous dataset.

Based on the additional information given by the company, we were able to write a small script that allows us to make a distinction between the different kinds of messages. In total there are around 7.340.000 Dutch messages created in 2006. 72% of them are automatically created, 23% manually and 5% semi-automatically.

The same steps of the approach were applied to this dataset. Again, we selected a set of tags which have a frequency of more than 1.000, and employed the same threshold values as described in the approach. Finally, we received the result displayed in figure 4.2.

4.3. Discussion of Results. When visually comparing the output of the two message types, we notice that the 2 generated lightweight ontologies contain different terms. This means that the tag usage between the two message types differs. Consequently, we will need to find a way to map the different partial results into a complete ontology.

We notice that we have captured other relations than merely broader/narrower or *a kind of* relations. For instance the relation between the tags *name of shop* and *baby*, can not really be considered as *a kind of* relation

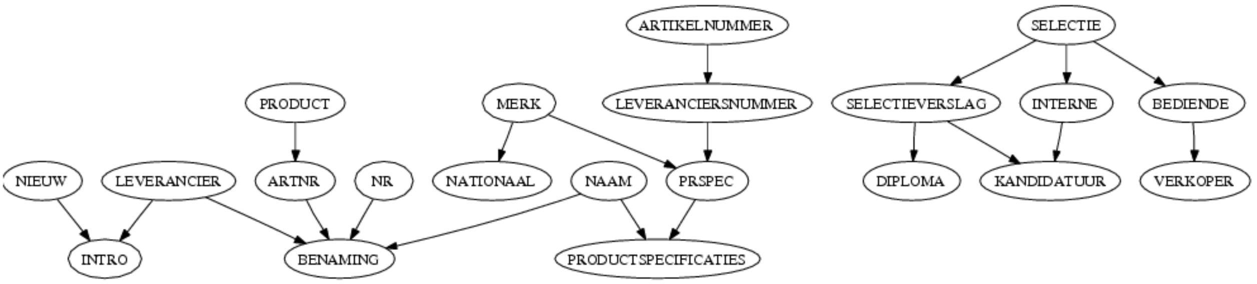


Fig. 4.2. *Partial results obtained from the message type "Notes" from the second dataset*

but more like a *is related to* relation. It provides more information regarding a stock item of the shop. Therefore, it would be interesting to find a way to capture these different kinds of relations and also check whether we may still apply transitive reduction.

We also observed that the graphs, as in figure 3.1, include some tags corresponding to the French language such as *article*, *bebe*, *magasin*, *piece*, *rechange*. When having a closer look at the data set, we noticed that there are some bilingual messages with bilingual tags. The tags can not be directly filtered from the database since there is no unique identifier. Looking at the results, we observed a pattern: the same tag relation exists between the Dutch and French tag pair e.g. in figure 3.1 (artikel, baby) and (article, bebe). We also observed this in the other results which are not visually included in this paper.

Tests with the Levenhstein metric, revealed that we can eliminate some French tags due to the close similarity among both languages e.g. *factuur* in Dutch and *facture* in French. In this way, the Levenshtein metric can reduce the pollution by French tags.

By applying our approach to these tags, we have reduced their tag's weaknesses as described in the first section. We now know with which other terms tags are mostly used together, for instance the tag *fout* is often used together with the tag *name_ of_ shop*. Pollution such as singular and plural tags is filtered out.

Since some parts of the obtained lightweight ontology are logically interpretable, we briefly verified the results by presenting them to the IT-director and the communication system's analyst of the company. They verified the results by looking at the visualizations and checking the tags in the communication system messaging system. They both confirmed that it reflects the company's terminology. Therefore, we concluded that the approach would be valuable to improve the communication among the employees. It visualizes how terms are often used together. When applying the approach on the tag dataset of every department, we should be able to compare the terminology of the different departments.

5. Possible Applications. Ontologies can be used to improve the communication in the company as motivated by [24, 6]. However, we believe that the methodology which we presented in this paper can be used for other applications than merely improving the communication among the employees in the company. The fact that the methodology is based (1) on the analysis of meta data or tags generated by the employees in the company and (2) the tagging process of the company under study is completely integrated with the actual business processes, generates a broad overview on the activities taken place over a certain time period.

As we will explain in the next paragraphs, we believe the visualization obtained from the approach could be used as a decision tool for management, follow-up tool for new terminology and as a tool for the creation of new teams.

5.1. Decision Management Tool. We believe that our methodology of building a visual lightweight corporate ontology from a folksonomy can be considered as a kind of business intelligence tool. Business intelligence aims at discovering interesting information based on analyzing the existing data in the company in order to improve the decision making process and generate a competitive advantage [5].

By observing figure 1, we noticed two remarkable relations. On the one hand, we saw that there exists a link between the *name of shop* (we renamed this tag to guarantee the anonymity of the company) and the tag *fout* or *mistake* in English. On the other hand, we found a relationship between the *name of shop* and the tags *Tongerl* and *Fil3965*. The tag *Tongerl* is used as the abbreviation for a Belgian city and *Fil3965* is the ID of one of the shops. The first mentioned relationship could be a signal that something is wrong and that the relationship between these tags should be further investigated. The latter one could indicate that the shop *Fil3965* has high sales revenue or high customer's complaints. By taking the time factor into account, these results could be compared over different time periods. Therefore, the approach presented in this paper might be an interesting tool for high-level managers in the company. High-level managers are more focused on higher level company's issues such as corporate strategy and are not always aware of all the things that are going on in the company. The visualization of the lightweight ontology obtained through our approach could support them in their daily work and help them in decision making. Therefore, we regard it as a kind of tool for decision making or a sort of add-on for an existing business intelligence tool.

5.2. Follow-up Tool for new Terminology. The proposed approach could be valuable as a follow-up tool for new corporate terminology. It reveals how new terms are utilized and interpreted. In the case of company acquisition, such an approach could be very interesting. When a company gets acquired by another company, the acquired company will have to apply new terminology to improve the communication process between both of them. Again, the time factor can be included in the process to evaluate and compare the results.

5.3. Creating Teams. When new teams have to be set up, the approach might be helpful to choose the most appropriate employees. This visualization shows how tags are combined with other ones. By selecting all the terms that are related to a certain word, the corresponding employees could be selected for the creation of a new team. Of course, social networking techniques [16] which can be used to cluster employees based on shared tags, can be used as an additional technique to find employees.

6. Discussion and Future Research. Next to briefly validating the approach by presenting the results to the IT-director and communication system's analyst of the company, we also discussed the possible applications of the approach. In their opinion, the first and third application benefit would be most interesting to their company. They even suggested a visual search tool as an additional application. Such as tool could be an extension of the suggested management tool. When the manager finds an interesting hierarchical relation or cluster, he should be able to click on it to retrieve the corresponding messages.

We plan to expand our tests to other message types to verify the applications which we deduced from our current results. In addition, we should set up focus groups with employees of the company where the results and the possible applications can be extensively discussed. The approach should be further extended and include more techniques and algorithms such as clustering techniques. In this way, more relations might be included in the ontology.

A threshold value that determines the minimal optimal frequency of a certain tag to be taken into account when applying our methodology should also be found.

When taking tags into account for business intelligence applications, the quality of the tags, becomes an important issue. Tagging does not restrict its users to use a predefined controlled vocabulary, they are free to use whatever tags or keywords they like. Since no control mechanism is included, there is no certitude regarding the quality of the tags. Therefore, metrics to automatically detect high quality tags becomes a real necessity.

Further, we will try to find a method to map the ontologies obtained by applying the approach to different message types. However, we believe a cost-benefit analysis should also be built-in in the approach to evaluate whether a more extended version of the ontology will generate the necessarily return on investment. Currently, the approach minimizes the human input and in this way a lightweight-ontology is cost-efficiently derived from the corporate folksonomy.

7. Conclusion. Companies need a corporate ontology because it can improve the communication among the employees. Since current ontology engineering techniques have some disadvantages, we proposed a new ontology engineering technique based on corporate folksonomies. It is a 6-step approach to turn a corporate folksonomy into a lightweight corporate ontology. By means of a corporate folksonomy, we applied our approach to an existing corporate folksonomy dataset. Based on a first small validation we concluded that the obtained lightweight ontology reflects the company's terminology and might help to improve the communication among the employees. We also deduced a number of possible applications for a company: decision tool for management, follow-up tool for new terminology and as a tool for the creation of new teams.

REFERENCES

[1] A. V. Aho, M. R. Garey, and J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J. Comput., 1 (1972), pp. 131–137.

[2] E. P. Bontas and C. Tempich, *Ontology engineering: A reality check*, in The 5th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE2006), R. Meersman, Z. Tari, et al., eds., vol. 4275 of LNCS, Montpellier, France, Nov 2006, Springer, pp. 836–854.

[3] C. V. Damme, S. Christiaens, and E. Vandijck, *Building an employee-driven crm ontology*, in Proceedings of the IADIS Multi Conference on Computer Science and Information Systems (MCCSIS): E-society2007, 2007.

[4] C. V. Damme, M. Hepp, and K. Siorpaes, *Folksontology: An integrated approach for turning folksonomies into ontologies*, in Proceedings of Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007), 2007, pp. 57–70.

[5] P. Davies, *Intelligence, Information Technology, and Information Warfare.*, Annual Review of Information Science and Technology (ARIST), 36 (2002), pp. 313–52.

[6] D. Fensel, *Ontologies: : A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer, 2003.

[7] M. S. Fox, *The tove project towards a common-sense model of the enterprise*, in Proceedings of IEA/AIE, London, UK, 1992, Springer-Verlag, pp. 25–34.

[8] W. Gale, K. Church, and D. Yarowsky, *Estimating upper and lower bounds on the performance of word-sense disambiguation programs*, in Proceedings of the 30th annual meeting on Association for Computational Linguistics, Association for Computational Linguistics Morristown, NJ, USA, 1992, pp. 249–256.

[9] S. Golder and B. A. Huberman, *Usage patterns of collaborative tagging systems*, Journal of Information Science 32(2), (2006), pp. 198–208.

[10] M. Guy and E. Tonkin, *Tidying up tags*, 2006.

[11] M. Hepp, *Possible ontologies: How reality constrains the development of relevant ontologies*, IEEE Internet Computing, 11 (2007), pp. 96–102.

[12] A. John and D. Seligmann, *Collaborative tagging and expertise in the enterprise*, in Proceedings of Collaborative Web Tagging Workshop at WWW2006, Edinburgh, UK, 2006.

[13] V. I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, Tech. Rep. 8, 1966.

[14] H. Luhn, *The automatic creation of literature abstracts, iBM J*, Res. Develop, 2 (1959), pp. 159–165.

[15] Y. Matsuo and M. Ishizuka, *Keyword Extraction from a Single Document Using Word Co-Occurrence Statistical Information*, INTERNATIONAL JOURNAL ON ARTIFICIAL INTELLIGENCE TOOLS, 13 (2004), pp. 157–170.

[16] P. Mika, *Ontologies are us: A unified model of social networks and semantics*, Web Semantics., 5 (2007), pp. 5–15.

[17] D. R. Millen, J. Feinberg, and B. Kerr, *Dogear: Social bookmarking in the enterprise*, in Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, 2006, ACM, pp. 111–120.

[18] F. Pereira, N. Tishby, and L. Lee, *Distributional clustering of English words*, in Proceedings of the 31st annual meeting on Association for Computational Linguistics, Association for Computational Linguistics Morristown, NJ, USA, 1993, pp. 183–190.

[19] M. Porter, *An algorithm for suffix stripping*, 14 (1980), pp. 130–137.

[20] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.

[21] P. Schmitz, *Inducing ontology from flickr tags*, in Proceedings of Collaborative Web Tagging Workshop at WWW2006, Edinburgh, UK, 2006.

[22] L. Specia and E. Motta, *Integrating folksonomies with the semantic web*, in Proceedings of the European Semantic Web Conference (ESWC2007), E. Franconi, M. Kifer, and W. May, eds., vol. 4519 of LNCS, Berlin Heidelberg, Germany, July 2007, Springer-Verlag, pp. 624–639.

[23] J. Surowiecki, *The Wisdom of Crowds*, Anchor, August 2005.

[24] M. Uschold and M. Grüninger, *Ontologies: principles, methods, and applications*, Knowledge Engineering Review, 11 (1996), pp. 93–155.

[25] M. Uschold and R. Jasper, *A framework for understanding and classifying ontology applications*, in Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden, 1999.

[26] M. Uschold, M. King, S. Moralee, and Y. Zorgios, *The enterprise ontology*, Knowledge Engineering Review, 13 (1998), pp. 31–89.

[27] T. Wang and B. Parsia, *Cropcircles: Topology sensitive visualization of owl class hierarchies*, in Proceedings of the International Semantic Web Conference (ISWC2006), I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, eds., vol. 4273 of LNCS, Springer-Verlag, November 2006, pp. 695–708.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: December 1st, 2007
Accepted: January 15th, 2008
Extended version received: June 30th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 303–314. http://www.scpe.org
ISSN 1895-1767
© 2008 SCPE

FUZZY CONSTRAINT-BASED SCHEMA MATCHING FORMULATION

ALSAYED ALGERGAWY, EIKE SCHALLEHN, AND GUNTER SAAKE*

Abstract. The deep Web has many challenges to be solved. Among them is schema matching. In this paper, we build a conceptual connection between the schema matching problem SMP and the fuzzy constraint optimization problem FCOP. In particular, we propose the use of the fuzzy constraint optimization problem as a framework to model and formalize the schema matching problem. By formalizing the SMP as a FCOP, we gain many benefits. First, we could express it as a combinatorial optimization problem with a set of soft constraints which are able to cope with uncertainty in schema matching. Second, the actual algorithm solution becomes independent of the concrete graph model, allowing us to change the model without affecting the algorithm by introducing a new level of abstraction. Moreover, we could discover complex matches easily. Finally, we could make a trade-off between schema matching performance aspects.

Key words: schema matching, constraint programming, fuzzy constraints, objective function

1. Introduction. The deep Web (also known as Deepnet or the hidden Web) refers to the World Wide Web content that is not a part of the surface Web. It is estimated that the deep Web is several orders of magnitude larger than the surface Web [4]. As the number of deep Web sources has been increasing as the efforts needed to enable users to explore and integrate these sources become essential. As a result software systems have been developed to open the deep Web to users. *Schema matching* is the core task of these systems.

Schema matching is the task of identifying semantic correspondences among elements of two or more schemas. It plays a central role in many data application scenarios [22, 17]: in *data integration*, to identify and characterize inter-schema relationships between multiple (heterogeneous) schemas; in *data warehousing*, to map data sources to a warehouse schema; in *E-business*, to help to map messages between different XML formats; in *the Semantic Web*, to establish semantic correspondences between concepts of different web sites ontologies; and in *data migration*, to migrate legacy data from multiple sources into a new one [10].

Due to the complexity of schema matching, it was mostly performed manually by a human expert. However, manual reconciliation tends to be a slow and inefficient process especially in large-scale and dynamic environments. Therefore, the need for automatic schema matching has become essential. Consequently, many schema matching systems have been developed for automating the process, such as Cupid [17], COMA/COMA++ [6, 1], LSD [8], Similarity Flooding [20], OntoBuilder [13], QOM [12], BTreeMatch [11], S-Match [14], and Spicy [3]. Manual semantic matching overcomes mismatches which exist in element names and also differentiates between differences of domains. Hence, we could assume that manual matching is a perfect process. On the other hand, automatic matching may carry with it a degree of uncertainty, as it is based on syntactic, rather than semantic, means. Furthermore, recently, there has been renewed interest in building database systems that handle uncertain data in a principled way [9]. Hence a short rant about the relationship between databases that manage uncertainty and data integration systems appears. Therefore, we should surf for a suitable model which is able to meet the above requirements.

A first step in discovering an effective and efficient way to solve any difficult problem such as schema matching is to construct a complete problem specification. A suitable and precise definition of schema matching is essential for investigating approaches to solve it. Schema matching has been extensively researched, and many matching systems have been developed. Some of these systems are rule-based [6, 17, 20] and others are learning-based [16, 7, 8]. However, formal specifications of problems being solved by these systems do not exist, or are partial. Little work is done towards schema matching problem formulation e.g. in [25, 23].

In the rule-based approaches, a graph is used to describe the state of a modeled system at a given time, and graph rules are used to describe the operations on the system's state. As a consequence in practice, using graph rules has a worst case complexity which is exponential to the size of the graph. Of course, an algorithm of exponential time complexity is unacceptable for serious system implementation. In general, to achieve acceptable performance it is inevitable to consequently exploit the special properties of both schemas to be matched. Beside that, there is a striking commonality in all rule-based approaches; they are all based on backtracking paradigms. Knowing that the overwhelming majority of theoretical as well as empirical studies on the optimization of backtracking algorithms is based on the context of constraint problem (CP), it is near

*Department of Computer Science, Magdeburg University, 39106 Magdeburg, Germany (Alshahat|eike|sake@ovgu.de).

to hand to open this knowledge base for schema matching algorithms by reformulating the schema matching problem as a CP [24, 18, 5].

To summarize, we are in a need to a framework which is able to face the following challenges:

1. *formalizing the schema matching problem*: Although many matching systems have been developed to solve the schema matching problem, but no complete work to address the formulation problem. Schema matching research mostly focuses on how well schema matching systems recognize correspondences. On the other hand, not enough research has been done on formal basics of the schema matching problem.
2. *trading-off between schema matching performance aspects*: The performance of a schema matching system comprises two equally important factors; namely *matching effectiveness* and *matching efficiency*. The effectiveness is concerned with the accuracy and the correctness of the match result while the efficiency is concerned with the system resources such as the response time of the match system. Recent schema matching systems report considerable effectiveness [6], however, the efficiency aspects remain a missing area and represent an open challenge for the schema matching community. Improving schema matching efficiency results in decreasing matching effectiveness, so a trade-off between the two aspects should be considered.
3. *dealing with uncertainty of schema matching:* Schema matching systems should be able to handle uncertainty arises during the matching process from different sources. Recently, there has been renewed interest in building database systems that handle uncertain data and its lineage in a principled way, so a short rant about the relationship between databases that manage uncertainty and lineage and data integration systems appears. In addition to, in order to fully automate the matching process, we make use of extractor tools which extract different data models and represent them as a common model. The extraction process brings errors and uncertainties to the matching process

In this paper, *we build a conceptual connection between the schema matching problem (SMP) and the fuzzy constraint optimization problem (FCOP).* On one hand, we consider schema matching as a new application of fuzzy constraints; on the other hand, we propose the use of the fuzzy constraint satisfaction problem as a new approach for schema matching. In particular, in this paper, we propose the use of the FCOP to formulate the SMP. However, our approach should be generic, i. e. have the ability to cope with different data models and be used for different application domains. Therefore, we first transform schemas to be matched into a common data model called rooted labeled graphs. Then we reformulate the graph matching problem as a constraint problem. There are many benefits behind this formulation. First, we gain direct access to the rich research findings in the CP area; instead of inventing new algorithms for graph matching from scratch. Second, the actual algorithm solution becomes independent of the concrete graph model, allowing us to change the model without affecting the algorithm by introducing a new level of abstraction. Third, formalizing the SMP as a FCOP facilitates handling uncertainty in the schema matching process. Finally, we could simply deal with simple and complex mappings.

The paper is organized as follows: Section 2 introduces necessary preliminaries. Our framework to unify schema matching is presented in Section 3 in order to illustrate the scope of this paper. Section 4 shows how to formulate the schema matching problem as a constraint problem. Section 5 describes the related work. The concluding remarks and ongoing future work are presented in Section 6.

2. Preliminaries.

This paper is based mainly on two existing bodies of research, namely graph theory [2] and constraint programming [24, 18, 5]. To keep this paper self-contained, we briefly present in this section the basic concepts of them.

2.1. Graph Model.

A schema is the description of the structure and the content of a model and consists of a set of related elements such as tables, columns, classes, or XML elements and attributes. There are many kinds of data models, such as relational model, object-oriented model, ER model, XML schema, etc. By schema structure and schema content, we mean its schema-based properties and its instance-based properties, respectively. In this subsection we present formally rooted (multi-)labeled directed graphs used to represent schemas to be matched as the internal common model.

A rooted labeled graph is a directed graph such that nodes and edges are associated with labels, and in which one node is labeled in a special way to distinguish it from the graph's other nodes. This special node is called the root of the graph. Without loss of generality, we shall assume that every node and edge is associated with at least one label: if some nodes (resp. edges) have no label, one can add an extra anonymous label that is associated with every node (resp. edge). More formally, we can define the labeled graph as follows:

Definition 2.1. *A Rooted Labeled Graph G is a 6-tuple G = (N_G, E_G, Lab_G, src, tar, l) where:*

- *$N_G = \{n_{root}, n_2, \ldots, n_n\}$ is a finite set of nodes, each of them is uniquely identified by an object identifier (OID), where n_{root} is the graph root.*
- *$E_G = \{(n_i, n_j)|n_i, n_j \in N_G\}$ is a finite set of edges, each edge represents the relationship between two nodes.*
- *Lab_G ={ Lab_{NG}, Lab_{EG} } is a finite set of node labels Lab_{NG} , and a finite set of edge labels Lab_{EG}. These labels are strings for describing the properties (features) of nodes and edges.*
- *src and tar: $E_G \mapsto N_G$ are two mappings (source and target), assigning a source and a target node to each edge (i. e. if $e = (n_i, n_j)$ then $src(e) = n_i$ and $tar(e) = n_j$).*
- *$l : N_G \cup E_G \mapsto Lab_G$ is a mapping label assigning a label from the given Lab_G to each node and each edge.*
- *$|N_G| = n$ is the graph size.*

Now that we have defined a concrete graph model, in the following subsection we present basics of constraint programming.

2.2. Constraint Programming. Many problems in computer science, most notably in Artificial Intelligence, can be interpreted as special cases of constraint problems. *Semantic schema matching is also an intelligence process which aims at mimicking the behavior of humans in finding semantic correspondences between schemas' elements. Therefore, constraint programming is a suitable scheme to represent the schema matching problem.*

Constraint programming is a generic framework for declarative description and effective solving for large, particularly combinatorial, problems. Not only it is based on a strong theoretical foundation but also it is attracting widespread commercial interest as well, in particular, in areas of modeling heterogeneous optimization and satisfaction problems. We, here, concentrate only on constraint satisfaction problems (CSPs) and present definitions for CSPs, constraints, and solutions for the CSPs.

Definition 2.2. *A Constraint Satisfaction Problem **P** is defined by a 3-tuple **P**=(X,D,C) where,*

- *$X = \{x_1, x_2, \ldots, x_n\}$ is a finite set of variables.*
- *$D = \{D_1, D_2, \ldots, D_n\}$ is a collection of finite domains. Each domain D_i is the set containing the possible values for the corresponding variable $x_i \in X$.*
- *$C = \{C_1, C_2, \ldots, C_m\}$ is a set of constraints on the variables of X.*

Definition 2.3. *A Constraint C_s on a set of variables $S = \{x_1, x_2, \ldots x_r\}$ is a pair $C_s = (S, R_s)$, where R_s is a subset on the product of these variables' domains: $R_s \subseteq D_1 \times \cdots \times D_r \rightarrow \{0, 1\}$.*

The number r of variables a constraint is defined upon is called arity of the constraint. The simplest type is the *unary constraint*, which restricts the value of a single variable. Of special interest are the constraints of arity two, called *binary constraints*. A constraint that is defined on more than two variables is called a *global constraint.*

Solving a CSP is finding assignments of values from the respective domains to the variables so that all constraints are satisfied.

Definition 2.4. *(Solution of a CSP) An assignment Λ is a solution of a CSP if it satisfies all the constraints of the problem, where the assignment Λ denotes an assignment of each variable x_i with the corresponding value a_i such that $x_i \in X$ and $a_i \in D_i$.*

Example 1. (Map Coloring) We want to color the regions of a map, shown in Fig. 2.1, in a way that no two adjacent regions have the same color. The actual problem is that only a certain limited number of colors is available. Let's we have four regions and only three colors. We now formulate this problem as $CSP = (X, D, C)$ where:

- $X = \{x_1, x_2, x_3, x_4\}$ represents the four regions,
- $D = \{D_1, D_2, D_3, D_4\}$ represents the domains of the variables such that $D_1 = D_2 = D_3 = D_4 = \{red, green, blue\}$, and
- $C = \{C_{(x1,x2)}, C_{(x1,x3)}, C_{(x1,x4)}, C_{(x2,x4)}, C_{(x3,x4)}\}$ represents the set of constraints must be satisfied such that $C_{(xi,xj)} = \{(vi, vj) \in Di \times Dj|vi \neq vj\}$.

As shown in Example 1, there are a number of solutions to the specified CSP. Any one of them is considered a solution to the problem. However, in the schema matching field, we do not only search for any solution but also the best one. The quality of solution is usually measured by an application dependent function called the objective function. The goal is to find such a solution that satisfies all the constraints and minimize or maximize the objective function. Such problems are referred to as constraint optimization problems (COP).

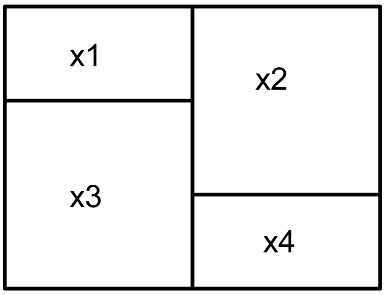


FIG. 2.1. *Map coloring example*

DEFINITION 2.5. *A Constraint Optimization Problem* $\boldsymbol{Q}$ *is defined by couple* $\boldsymbol{Q}$ *=(P,g) such that* $\boldsymbol{P}$ *is a CSP and* $g : D_1 \times \cdots \times D_n \rightarrow [0,1]$ *is an objective function that maps each solution tuple into a value.*

Example 2. (Traveling Salesman) The traveling salesman problem is to find the shortest closed path by which city out of a set of n cities is visited once and only once.

While powerful, both CSP and COP present some limitations. In particular, all constraints are considered mandatory. In many real-world problems, such as the schema matching problem, there are constraints that could be violated in solutions without causing such solutions to be unacceptable. If these constraints are treated as mandatory, this often causes problems to be unsolved. If these constraints are ignored, solutions of bad quality are found. *This is a motivation to extend the CSP scheme and make use of soft constraints.* A way to circumvent inconsistent constraints problems is to make them fuzzy [15]. The idea is to associate fuzzy values with the elements of the constraints, and combine them in a reasonable way.

A constrain, as defined before, is usually defined as a pair consisting of a set of variables and a relation on these variables. This definition gives us the availability to model different types of uncertainty in schema matching. In [9], authors identify different sources for uncertainty in data integration. Uncertainty in semantic mappings between data sources can be modeled by exploiting fuzzy relations while other sources of uncertainty can be modeled by making the variable set a fuzzy set. In this paper, we take the first one into account while the other sources are left for our ongoing work.

DEFINITION 2.6. *(Fuzzy Constraint) A Fuzzy Constraint* C_μ *on a set of variables* $S = \{x_1, x_2, \ldots, x_r\}$ *is a pair* $C_\mu = (S, R_\mu)$*, where the fuzzy relation* R_μ*, defined by* $\mu_R : \prod_{xi \in var(C)} D_i \mapsto [0,1]$ *where* μ_R *is the membership function indicating to what extent a tuple v satisfies* C_μ*.*

- $\mu_R(v) = 1$ *means v totally satisfies* C_μ*,*
- $\mu_R(v) = 0$ *means v totally violates* C_μ*, while*
- $0 < \mu_R(v) < 1$ *means v partially satisfies* C_μ*.*

DEFINITION 2.7. *A Fuzzy Constraint* C_μ *on a set of variables* $S = \{x_1, x_2, \ldots x_r\}$ *is a pair* $C_\mu = (S, R_\mu)$*, where the fuzzy relation* R_μ*, defined by* $\mu_R : \prod_{x_i \in var(C)} D_i \rightarrow [0,1]$ *where* μ_R *is the membership function indicating to what extent a tuple v satisfies* C_μ*.*

- $\mu_R(v) = 1$ *means v totaly satisfies* C_μ*,*
- $\mu_R(v) = 0$ *means v totaly violates* C_μ*, while*
- $0 < \mu_R(v) < 1$ *means v partially satisfies* C_μ*.*

DEFINITION 2.8. *A Fuzzy Constraint Optimization Problem* $\boldsymbol{Q}_\mu$ *is a 4-tuple* $\boldsymbol{Q}_\mu$*= (X, D, Cμ, g) where X is a list of variables, D is a list of domains of possible values for the variables,* C_μ *is a list of fuzzy constraints each of them referring to some of the given variables, and g is an objective function to be optimized.*

In the following section we shed the light on our schema matching framework to determine the scope of schema matching understanding.

3. A unified schema matching framework. Each of the existing schema matching systems deals with the schema matching problem from its point of view. As a result the need to a generic framework that unifies the solution of this intricate problem independent on the domain of schemas to be matched and independent on the model representations becomes essential. To this end, we suggest the following general phases to address the schema matching problem. Figure 2 shows these phases with the main scope of this paper. The four different phases are:

- importing schemas to be matched; *TransMat* Phase,

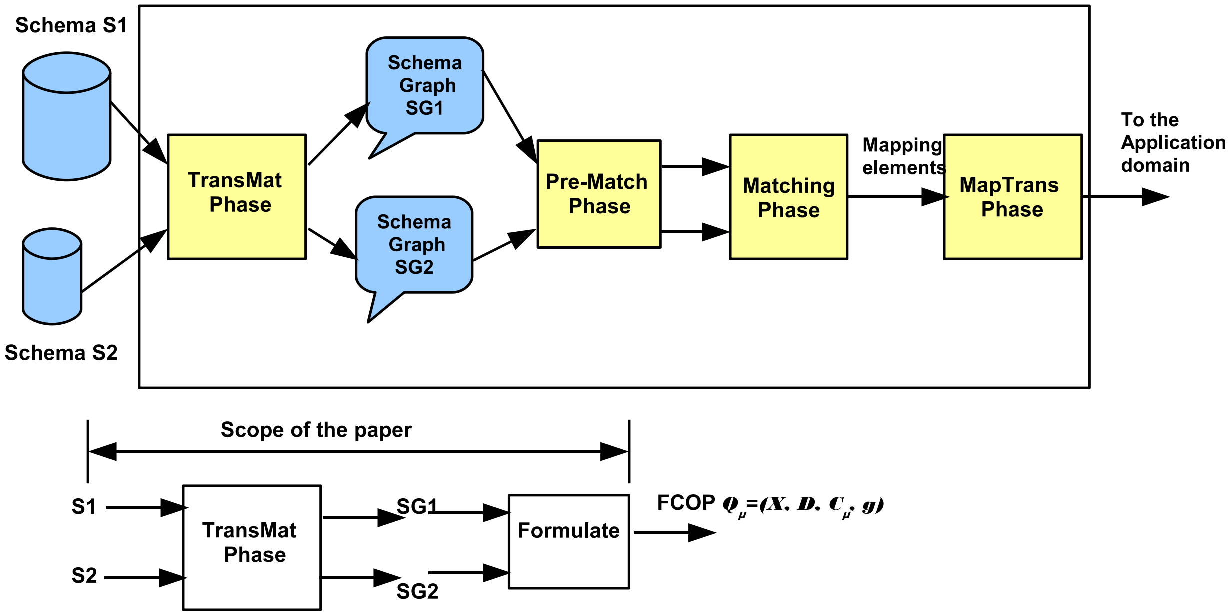


Fig. 3.1. *Matching Process Phases*

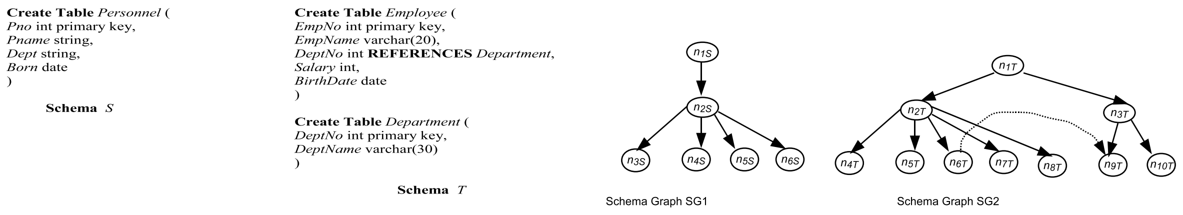
- identifying elements to be matched; *Pr-matching* Phase,
- applying the matching algorithms; *Matching* Phase, and
- exporting the match result; *MapTrans* Phase.

In the following subsection we introduce a framework for defining different data models and how to transform them into schema graphs. This part follows the same procedure found in [25] to show that different data models could be represented by schema graphs.

3.1. Schema Graph. To make the matching process a more generic process, schemas to be matched should be represented internally by a common representation. This uniform representation reduces the complexity of the matching process by not having to cope with different representations. By developing such import tools, schema match implementation can be applied to schemas of any data model such as *SQL*, *XML*, *UML*, and etc. Therefore, the first step in our approach is to transform schemas to be matched into a common model in order to apply matching algorithms. We make use of rooted labeled graphs as the internal model. We call this phase *TransMat*; Transformation for Matching process.

In general, to represent schemas and data instances, starting from the root, the schema is partitioned into relations and further down into attributes and instances. In particular, to represent relational schemas, XML schemas, etc. as rooted labeled graphs, independently of the specific source format, we benefit from the rules found in [25, 21]. These rules are rewritten as follows:

- Every prepared matching object in a schema such as the schema, relations, elements, attributes etc. is represented by a node, such that the schema itself is represented by the root node. Let schema S consist of m elements ($elem$), then
$$\forall\ elem \in S\ \exists\ n_i \in N_G \wedge S \mapsto n_{root},\ s.t.\ 1 \leq i \leq m$$
- The features of the prepared matching object are represented by node labels Lab_{NG}. Let features ($featS$) be the property set of an element ($elem$), then
$$\forall\ feat \in featS\ \exists\ Lab \in Lab_{NG}$$
- The relationship between two prepared matching objects is represented by an edge. Let the relationships between schema elements be ($relS$), then
$$\forall\ rel \in relS\ \exists\ e(n_i,\ n_j) \in E_G \text{ s. t. } src(e) = n_i \in N_G \wedge tar(e) = n_j \in N_G$$
- The properties of the relationship between prepared objects are represented by edge labels Lab_{EG}. Let features $rfeatS$ be the property set of a relationship rel, then,
$$\forall\ rfeat \in rfeatS\ \exists\ Lab \in Lab_{EG}$$



```
Create Table Personnel (
Pno int primary key,
Pname string,
Dept string,
Born date
)

        Schema S
```

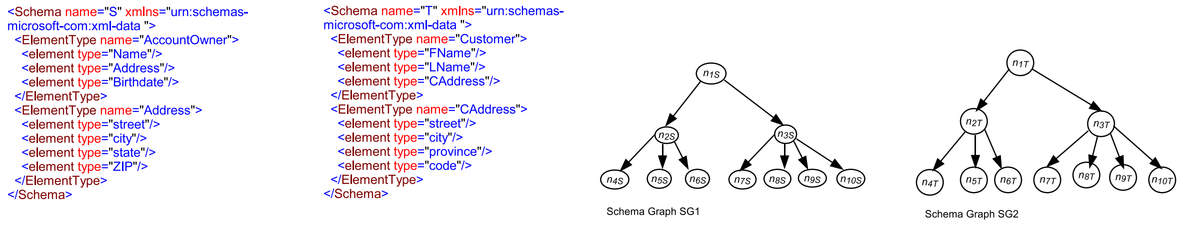
```
Create Table Employee (
EmpNo int primary key,
EmpName varchar(20),
DeptNo int REFERENCES Department,
Salary int,
BirthDate date
)

Create Table Department (
DeptNo int primary key,
DeptName varchar(30)
)

                Schema T
```

(a) Two relational schemas (b) Schema graphs

Fig. 3.2. *Two Relational Schemas & their Schema Graphs (without labels)*



```
<Schema name="S" xmlns="urn:schemas-
microsoft-com:xml-data ">
  <ElementType name="AccountOwner">
    <element type="Name"/>
    <element type="Address"/>
    <element type="Birthdate"/>
  </ElementType>
  <ElementType name="Address">
    <element type="street"/>
    <element type="city"/>
    <element type="state"/>
    <element type="ZIP"/>
  </ElementType>
</Schema>
```

```
<Schema name="T" xmlns="urn:schemas-
microsoft-com:xml-data ">
  <ElementType name="Customer">
    <element type="FName"/>
    <element type="LName"/>
    <element type="CAddress"/>
  </ElementType>
  <ElementType name="CAddress">
    <element type="street"/>
    <element type="city"/>
    <element type="province"/>
    <element type="code"/>
  </ElementType>
</Schema>
```

(a) Two XML schemas (b) Schema graphs

Fig. 3.3. *Two XML schemas & their schema graphs (without labels*

The following two examples illustrate that how these rules can be applied to different data models in order to make our approach a more generic approach.

Example 3. *(Relational Database Schemas)* Consider schemas S and T depicted in Fig. 3.2(a) (from [20]). The elements of S and T are tables and attributes. Applying the above rules, the two schemas *Schema* S and *Schema* T are represented by *SG1* and *SG2* respectively, such that $SG1 = (N_{GS}, E_{GS}, Lab_{GS}, src_S, tar_S, l_S)$, where

$$N_{GS} = \{n_{1S}, n_{2S}, n_{3S}, n_{4S}, n_{5S}, n_{6S}\}, \quad E_{GS} = \{e_{1-2}, e_{2-3}, e_{2-4}, e_{2-5}, e_{2-6}\},$$
$$Lab_{GS} = Lab_{NS} \cup Lab_{ES} = \{name, type, data\ type\} \cup \{part\text{-}of, associate\},$$

src_S, tar_S, l_S are mappings such that $src_S(e_{1-2}) = n_{1S}$, $tar_S(e_{2-3}) = n_{3S}$ and $l_S(e_{1-2}) = part\text{-}of$. Figure 3.2(b) shows only the nodes and edges of the schema graphs (SG2 can be defined similarly).

In this example, we exploit different features of matching objects such as name, datatype, and type. These features are represented as nodes' labels. These features shall be the input parameters to the next phase. For example, the name of a matching object in $SG1$ will be used to measure linguistic similarity between it and another matching object from $SG2$, its datatype is to measure datatype compatibility, and its type is used to determine semantic relationships. However, our approach is flexible in the sense that it is able to exploit more features as needed. Moreover, in this example, we exploit one structural feature "part-of" to represent structural relationships between nodes at different levels. Other structural features e.g. association relationship, that is a structural relationship specifying both nodes are conceptually at the same level, is represented between keys. One association relationship is represented in Fig. 3.2(b) between the nodes n_{6T} and n_{9T} to specify a key/foreign key relation. Visually, association edges are represented as dashed lines.

Example 4. (XML Schemas) This example that we discuss illustrates how our unified schema matching framework copes with different choices of the models to be matched. Now consider two XML schemas in Fig. 3.3(a) (from [25]). The schemas are specified using the XML language deployed on the website biztalk.org designed for electronic documents used in e-business. The schema graphs (without labels) of these schemas are shown in Fig. 3.3(b). The labels of nodes and edges are the same as Example 3.

Examples 3 and 4 illustrate that using Trans-Mat phase aims at matching different schema models. The matching algorithm (Matching Phase) does not have to deal with a large number of different models. The matching algorithm only deals with the internal representation. So far, recent schema matching systems directly determine semantic correspondences between two schemas elements as a graph matching problem. In this paper, we extend the internal representation, schema graphs, and reformulate the graph matching problem as a fuzzy constraint optimization problem.

4. Schema Matching as a FCOP.

4.1. Schema Matching as Graph Matching. Schemas to be matched are transformed into rooted labeled graphs and, hence, the schema matching problem is converted into graph matching. There are two types of graph matching: *graph isomorphism* and *graph homomorphism*. In general, a match of one graph into another is given by a *graph morphism*, which is a mapping of one graph's object sets into the other's, with some restrictions to preserve the graph's structure and its typing information.

Definition 4.1. *A Graph Morphism $\phi : SG1 \rightarrow SG2$ between two schema graphs*

$$SG1 = (N_{GS}, E_{GS}, Lab_{GS}, src_S, tar_S, l_S) \text{ and } SG2 = (N_{GT}, E_{GT}, Lab_{GT}, src_T, tar_T, l_T)$$

is a pair of mappings $\phi = (\phi_N, \phi_E)$ such that $\phi_N : N_{GS} \rightarrow N_{GT}$ (ϕ_N is a node mapping function) and $\phi_E : E_{GS} \rightarrow E_{GT}$ (ϕ_E is an edge mapping function) and the following restrictions apply:

1. $\forall n \in N_{GS} \; \exists \; l_S(n) = l_T(\phi_N(n))$
2. $\forall e \in E_{GS} \; \exists \; l_S(e) = l_T(\phi_E(e))$
3. $\forall e \in E_{GS} \; \exists$ *a path* $p' \in N_{GT} \times E_{GT}$ *such that* $p' = \phi_E(e)$ *and* $\phi_N(src_S(e)) = src_T(\phi_E(e)) \wedge \phi_N(tar_S(e)) = tar_T(\phi_E(e))$.

The first two conditions preserve both nodes and edges labeling information, while the third condition preserves graph's structure. Graph matching is an isomorphic matching problem when $|N_{GS}| = |N_{GT}|$ otherwise it is homomorphic. Obviously, the schema matching problem is a homomorphic problem.

Example 5. For the two relational schemas depicted in Fig.3.2(a) and its associated schema graphs shown in Fig.3.2(b), the schema matching problem between schema S and schema T is converted into a homomorphic graph matching problem between $SG1$ and $SG2$.

Graph matching is considered to be one of the most complex problems in computer science. Its complexity is due to two major problems. The first problem is the computational complexity of graph matching. The time required by backtracking in search tree algorithms may in the worst case become exponential in the size of the graph. Graph homomorphism has been proven to be NP-complete problem [19]. The second problem is the fact that all of the algorithms for graph matching mentioned so far can only be applied to two graphs at a time. Therefore, if there are more than two schemas that must be matched, then the conventional graph matching algorithms must be applied to each pair sequentially. For applications dealing with large databases, this may be prohibitive. Hence, choosing graph matching as platform to solve the schema matching problem may be effective process but inefficient. Therefore, we propose transforming graph homomorphism into a FCOP.

Now that we have defined a graph model and its homomorphism, let us consider how to construct a FCOP out of a given graph matching problem.

4.2. Graph Matching as a FCOP. In the schema matching problem, we are trying to find a mapping between the elements of two schemas. Multiple conditions should be applied to make these mappings valid solutions to the matching problem, and some objective functions are to be optimized to select the best mappings among matching result. The analogy to constraint problem is quite obvious: here we make a mapping between two sets, namely between a set of variables and a set of domains, where some conditions should be satisfied. So basically, what we have to do to obtain an equivalent constraint problem CP for a given schema matching problem (knowing that schemas to be matched are transformed into schema graphs) are:

1. take objects of one schema graph to be matched as the CP's set of variables,
2. take objects of other schema graphs to be matched as the variables' domain,
3. find a proper translation of the conditions that apply to schema matching into a set of fuzzy constraints, and
4. form objective functions to be optimized.

We have defined the schema matching problem as a graph matching homomorphism ϕ. We now proceed by formalizing the problem ϕ as a FCOP problem $Q_\mu = (X, D, C_\mu, g)$. To construct a FCOP out of this problem, we follow the above rules. Through these rules, we take the two relational database schemas shown in Fig. 3.2(a) and its associated schema graphs shown in Fig. 3.2(b) as an example, taking into account that $|N_{GS}(=6)| < |N_{GT}(=10)|$ as follows:

- The set of variables X is given by $X = N_{GS} \bigcup E_{GS}$ where the variables from N_{GS} are called node variables X_N and from E_{GS} are called edge variables X_E
$$X = X_N \bigcup X_E$$
$$= \{x_{n1}, x_{n2}, x_{n3}, x_{n4}, x_{n5}, x_{n6}\} \bigcup \{x_{e1-2}, x_{e2-3}, x_{e2-4}, x_{e2-5}, x_{e2-6}\}$$
- The set of domain D is given by $D = N_{GT} \bigcup E_{GT}$, where the domains from N_{GT} are called node domains D_N and from E_{GT} are called edge domains D_E,
$= \{D_{n1}, D_{n2}, D_{n3}, D_{n4}, D_{n5}, D_{n6}\} \bigcup \{D_{e1-2}, D_{e2-3}, D_{e2-4}, D_{e2-5}, D_{e2-6}\}$ where $D_{n1} = D_{n2} = D_{n3} = D_{n4} = D_{n5} = D_{n6} =$
$\{n_{1T}, n_{2T}, n_{3T}, n_{4T}, n_{5T}, n_{6T}, n_{7T}, n_{8T}, n_{9T}, n_{10T}\}$ (i. e. the node domain contains all the second schema graph nodes) and $D_{e1-2} = D_{e2-3} = D_{e2-4} = D_{e2-5} = D_{e2-6} =$
$\{e_{1-2T}, e_{1-3T}, e_{2-4T}, \ldots, p_{1-2-4T}, \ldots\}$ (i. e. the edge domain contains all the available edges and paths in the second schema graph) (the edge e_{1-2} reads the edge extends between the two nodes n_1 and n_2 such that $e_{1-2} = e(n_1, n_2)$).

Using this formalization enables us to deal with holistic matching. This can be achieved by taking the objects of one schema as the variable set, while the objects of other schemas as the variable's domain. Let we have n schemas which are transformed into schema graphs $SG1, SG2, \ldots, SGn$ then $X = X_N \bigcup X_E$, $D_N = \sum_{i=2}^{n} D_{Ni}$, $D_E = \sum_{i=2}^{n} D_{Ei}$. Another benefit behind this approach is that our approach is able to discover complex matches of types *1:n* and *n:1* very easily. This can be achieved by allowing a value may have multiple values from its corresponding domain and a value may be assigned to multiple variables.

In the following subsections, we demonstrate how to construct both constraints and objective functions in order to obtain a complete problem definition.

4.3. Constraint Construction. The exploited constraints should reflect the goals of schema matching. Schema matching based only on schema element properties has been attempted. However, it does not provide any facility to optimize matching. Furthermore, additional constraint information, such as semantic relationships and other domain constraints, is not included and schemas may not completely capture the semantics of data they describe. Therefore, in order to improve performance and correctness of matching, additional information should be included. In this paper, we are concerned with both syntactic and semantic matching. Therefore, we shall classify constraints that should be incorporated in the CP model into*: syntactic constraints* and *semantic constraints*. In the following, we consider only the constraints construction while the fuzzy relations of fuzzy constraint are not consider since it depends on the application domain. For example, as shown below, domain constraints are crisp constraints, i. e. $\mu_C(v) = 1$, while the structural constraints are soft constraints with different degree of satisfaction.

4.3.1. Syntactic Constraints.

1. ***Domain Constraints***: It states that a node variable must be assign a value or a set of values from its corresponding node domain, and an edge variable must be assigned a value from its corresponding edge domain. That is $\forall x_{ni} \in X_N$ and $x_{ej} \in X_E \exists$ a unary constraint$C^{dom}_{\mu(x_{ni})}$ and $C^{dom}_{\mu(x_{ei})}$ ensuring domain consistency of the match where,
$$C^{dom}_{\mu(x_{ni})} = \{d_i \in D_{Ni}\},$$
$$C^{dom}_{\mu(x_{ei})} = \{d_i \in D_{Ei}\}$$
2. ***Structural Constraints***: There are many structural relationships between schema graph nodes such as:
   - *Edge Constraint*: It states that if an edge exists between two variable nodes, then an edge (or path) should exist between their corresponding images. That is $\forall x_{ei} \in X_E$ and its source and target nodes are x_{ns} and $x_{nt} \in X \exists$ two binary constraints $C^{src}_{\mu(x_{ei}, x_{ns})}$ and $C^{tar}_{\mu(x_{ei}, x_{nt})}$ representing the structural behavior of matching, where:
   $$C^{src}_{\mu(x_{ei}, x_{ns})} = \{(d_i, d_j) \in D_E \times D_N | src(d_i) = d_j\}$$
   $$C^{tar}_{(x_{ei}, x_{nt})} = \{(d_i, d_j) \in D_E \times D_N | tar(d_i) = d_j\}$$
   - $\forall$ two variables nodes x_{ni} and $x_{nj} \in X \exists$ a set of binary constraints describing the hierarchical relationships between schema graph nodes as follows:

(a) *Parent Constraint* $C^{parent}_{\mu(x_{ni},x_{nj})}$ representing the structural behavior of parent relationship, where

$$C^{parent}_{\mu(x_{ni},x_{nj})} = \{(d_i, d_j) \in D_N \times D_N \mid \exists e(d_i, d_j) \ s.t. \ src(e) = d_i\}$$

(b) *Child Constraint* $C^{child}_{\mu(x_{ni},x_{nj})}$ representing the structural behavior of child relationship, where

$$C^{child}_{\mu(x_{ni},x_{nj})} = \{(d_i, d_j) \in D_N \times D_N \mid \exists e(d_i, d_j) \ s.t. \ tar(e) = d_j\}$$

(c) *Sibling Constraint* $C^{sibl}_{\mu(x_{ni},x_{nj})}$ representing the structural behavior of sibling relationship, where

$$C^{sibl}_{\mu(x_{ni},x_{nj})} = \{(d_i, d_j) \in D_N \times D_N \mid \exists d_n \ s.t. \ parent(d_n, d_i) \wedge parent(d_n, d_j)\}$$

4.3.2. Semantic Constraints. The first constraint type considers only the structural and hierarchical relationships between schema graph nodes. In order to capture the other features of schema graph nodes such as the semantic feature we make use of the following constraint.

1. Label Constraints: $\forall x_{ni} \in X_N$ and $\forall x_{ei} \in X_E \exists$ a unary constraint $C^{Lab}_{\mu(xni)}$ and $C^{Lab}_{\mu(xei)}$ ensuring the semantics of the predicates in the schema such that:

$$C^{Lab}_{\mu(x_{ni})} = \{dj \in DN | lsim(lS(x_{ni}), lT(d_j)) \geq t\}$$
$$C^{Lab}_{\mu(x_{ei})} = \{dj \in DE | lsim(lS(x_{ei}), lT(d_j)) \geq t\}$$

where $lsim$ is a linguistic similarity function determining the semantic similarity between nodes/edges labels and t is a predefined threshold.

The above syntactic and semantic constraints are by no means the contextual relationships between elements. Other kinds of domain knowledge can also be represented through constraints. Moreover, each constraint is associated with a membership function $\mu(v) \in [0, 1]$ to indicate to what extent the constraint should be satisfied. If $\mu(v) = 0$, this means v totally violates the constraint and $\mu(v) = 1$ means v totally satisfies it. Constraints restrict the search space for the matching problem so may benefit the efficiency of the search process. On the other hand, if too complex, constraints introduce additional computational complexity to the problem solver.

4.4. Objective Function Construction. The objective function is the function associated with an optimization process which determines how good a solution is and depends on the object parameters. The objective function constitutes the implementation of the problem to be solved. The input parameters are the object parameters. The output is the objective value representing the evaluation/quality of the individual. In the schema matching problem, the objective function simulates human reasoning on similarity between schema graph objects.

In this framework, we should consider two function components which constitute the objective function. The first is called cost function f_{cost} which determines the cost of a set constraint over variables. The second is called energy function f_{energy} which maps every possible variable assignment to a cost. Then, the objective function could be expressed as follows:

$$g = \text{mis}|\max(\sum\nolimits_{\text{set of constraints}} f_{\text{cost}} + \sum\nolimits_{\text{set of assignment}} f_{\text{energy}})$$

5. Related Work. Schema matching is a fundamental process in many domains dealing with shared data such as data integration, data warehouse, E-commerce, semantic query processing, and the web semantics. Matching solutions were developed using different kind of heuristics, but usually without prior formal definition of the problem they are solving. Although many matching systems, such as Cupid [17], COMA/COMA++ [6, 1], LSD [8], Similarity Flooding [20], OntoBuilder [13], QOM [12], BTreeMatch [11], S-Match [14], and Spicy [3], have been developed and different approaches have been proposed to solve the schema matching problem, but no complete work to address the formulation problem. Schema matching research mostly focuses on how well schema matching systems recognize corresponding schema elements. On the other hand, not enough research has been done on formal basics of the schema matching problem.

Most of the existing work [22] define match as a function that takes two schemas (models) as input, may be in the presence of auxiliary information sources such as user feedback and previous mappings, and produces a mapping as output. A schema consists of a set of related elements such as tables, columns, classes, or XML elements and attributes. A mapping is a set of mapping elements specifying the matching schema elements together. Each mapping element is specified by 4-tuple element $\langle ID, S_i^1, S_j^2, R\rangle$ where ID is an identifier for the mapping element that matches between the element S_i^1 of the first schema and the element S_j^2 of the second one and R indicates the similarity value between 0 and 1. The value of 0 means strong dissimilarity while the value of 1 means strong similarity. But, in general, a mapping element indicates that certain element(s) of schema $S1$ are related to certain element(s) of schema $S2$. Each mapping element can have an associated mapping expression which specifies how the two elements (or more) are related. Schema matching is considered only with identifying the mappings not determining the associated expressions.

In the work of A. Doan [7], they formalize the schema matching problem as four different problems:

1. *The basic 1-1 Matching*; given two schemas S and T (representations), for each element s of S, find the most semantically similar element t of T, utilizing all available information. This problem is often referred as a one-to-one matching problem, because it matches each element s with a single element. For example, the ⟨*ID1, S.Address, T.CAddress, 0.8*⟩ mapping element indicates that there a mapping between the element S.*Address* of schema S and the element T.*CAddress* of schema T with a degree of similarity 0.8.
2. *Matching for Data Integration*; given source schemas *S1, S2,. . . ,Sn* and mediated schema T, for each element s of Si find the most similar element t of T.
3. *Complex Matching*; let S and T be two data representations. Let *O ={O1, O2,. . . ,Ok}* be a set of operators that can be applied to the elements of T according to a set of rules R to Figure 2: Matching Function construct formulas. For each element s of S, find the most similar element t, where t can be either an element of T or a formula from the elements of T, using O and R.
4. *Matching for Taxonomies*; given two taxonomies of concepts S and T, for each concept node s of S, find the most similar concept node of T.

For each of these problems, Doan shows input information, solution output, and the evaluation of a solution output. In general, the input to a problem can include any type of knowledge about the schemas to be matched and their domains such as schema information, instance data, previous matchings, domain constraints, and user feedback.

Zhang and et. el. [25] formulate the schema matching problem as a combinatorial optimization problem. The authors cast the schema matching problem into a multi-labeled graph matching problem. The authors propose a meta-meta model of schema: multi-labeled graph model, which views schemas as finite structures over the specific signatures. Based on this multi-labeled schema, they propose a multi-labeled graph model, which is an instance of multi-label schema, to describe various schemas, where each node and edge can be associated with a set of labels describing its properties. Then they construct a generic graph similarity measure based on the contrast model and propose an optimization function to compare two multi-labeled graphs. Using the greedy algorithm, they design an optimization algorithm to solve the multi-labeled graph matching problem.

Gal and et al. [13] propose a fuzzy framework to model the uncertainty of the schema matching process outcome. The framework aims at identifying and analyzing factors that impact the effectiveness of schema matching algorithms by reducing the uncertainty of existing algorithms. To specify their belief in the mapping quality, the authors associate a confidence measure with any mapping among attributes' sets. They use the framework to define the monotonicity property as a desired property of the schema matching problem, so one can safely interpret a high confidence measure as a good semantic mapping.

The recent work for [23] introduces a formal specification for the XML matching problem. The authors define the ingredients of the XML schema matching problem using constraint logic programming. Matching problems can be defined through variables, variable domains, constraints and an objective function. They distinguish between the constraint satisfaction problem and constraint optimization problem and show that the optimization problem is more suitable for the schema matching problem. They make use of combination of clustering methods and the branch and bound algorithm to solve the schema matching problem.

In our formulation approach, we have some common and distinct features with the other related work. The common features include transforming schemas to be matched into schema graphs, i. e. rooted labeled graphs, and making use of the constraint programming as a framework to extend the graph matching problem into a constraint optimization problem. However, our approach introduces distinctly the use of fuzzy constraint in order to reflect the nature of the schema matching problem. As well as the use of the fuzzy constraint enables us to model uncertainty in the schema matching process.

6. Summary and Future Work. In this paper, we have investigated an intricate problem; the schema matching problem. In particular, we have introduced a fuzzy constraint-based framework to model the schema matching problem. To this end, we build a conceptual connection between the schema matching problem and fuzzy constraint optimization problem. On one hand, we consider schema matching as a new application of fuzzy constraint optimization, and on the other hand we propose the use of fuzzy constraint optimization as a new approach for schema matching.

Our proposed approach is a generic framework which has the feature to deal with different schema representations by transforming the schema matching problem into graph matching. Instead of solving the graph matching problem which has been proven to be an NP-complete problem, we reformulate it as a constraint problem. We have identified two types of constraints syntactic and semantic to ensure match semantics. As well as, we make use of the fuzzy constraints in order to enable us modeling uncertainty in the schema matching process. We also shed light on how to construct objective functions.

The main benefit of this approach is that we gain direct access to the rich research findings in the CP area; instead of inventing new algorithms for graph matching from scratch. Another important advantage is that the actual algorithm solution becomes independent of the concrete graph model, allowing us to change the model without affecting the algorithm by introducing a new level of abstraction.

Understanding the schema matching problem is considered the first step towards an effective and efficient solution for the problem. In our ongoing work, we will exploit constraint solver algorithms to reach our goal.

REFERENCES

[1] D. Aumueller, H. H. Do, S. Massmann, and E. Rahm, *Schema and ontology matching with COMA++*, in SIGMOD Conference, 2005, pp. 906–908.

[2] R. Babakrishnan and K.Ranganathan, *A textbook of graph theory*, Spring Verlag, 1999.

[3] A. Bonifati, G. Mecca, A. Pappalardo, and S. Raunich, *The spicy project: A new approach to data matching*, in SEBD, Turkey, 2006.

[4] S. C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, *Structured databases on the web: Observations and implications*, SIGMOD Record, 33 (2004), pp. 61–70.

[5] R. Dechter, *Constraint Processing*, Morgan Kaufmann, 2003.

[6] H. H. Do and E. Rahm, *COMA- a system for flexible combination of schema matching approaches*, in VLDB 2002, 2002, pp. 610–621.

[7] A. Doan, *Learning to map between structured representations of datag*, in Ph.D Thesis, Washington University, 2002.

[8] A. Doan, P. Domingos, and A. Halevy, *Reconciling schemas of disparate data sources: A machine-learning approach*, SIGMOD, (2001), pp. 509–520.

[9] X. Dong, A. Halevy, and C. Yu, *Data integration with uncertainty*, in VLDB'07, 2007, pp. 687–698.

[10] C. Drumm, M. Schmitt, H.-H. Do, and E. Rahm, *Quickmig - automatic schema matching for data migration projects*, in Proc. ACM CIKM07, Portugal, 2007.

[11] F. Duchateau, Z. Bellahsene, and M. Roche, *An indexing structure for automatic schema matching*, in SMDB Workshop, Turkey, 2007.

[12] M. Ehrig and S. Staab, *QOM- quick ontology mapping*, in International Semantic Web Conference, 2004, pp. 683–697.

[13] A. Gal, A. Tavor, A. Trombetta, and D. Montesi, *A framework for modeling and evaluating automatic semantic reconciliation*, VLDB Journal, 14 (2005), pp. 50–67.

[14] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, *Semantic matching: Algorithms and implementation*, Journal on Data Semantics, IX (2007).

[15] H. W. Guesgen and A. Philpott, *Heuristics for fuzzy constraint satisfaction*, in ANNES '95, 1995, pp. 132–135.

[16] W. Li and C. Clifton, *Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks*, Data and Knowledge Engineering, 33 (2000), pp. 49–84.

[17] J. Madhavan, P. A. Bernstein, and E. Rahm, *Generic schema matching with cupid*, in VLDB 2001, Roma, Italy, 2001, pp. 49–58.

[18] K. Marriott and P. Stuckey, *Programming with Constraints: An Introduction*, MIT Press, 1998.

[19] S. Medasani, R. Krishnapuram, and Y. Choi, *Graph matching by relaxation of fuzzy assignments*, IEEE Trans. on Fuzzy Systems, 9 (2001), pp. 173–182.

[20] S. Melnik, H. Garcia-Molina, and E. Rahm, *Similarity flooding: A versatile graph matching algorithm and its application to schema matching*, in ICDE'02, 2002.

[21] L. Palopoli, D. Rossaci, G. Terracina, and D. Ursino, *A graph-based approach for extracting terminological properties from information sources with heterogeneous formats*, Knowledge and Information Systems, 8 (2005), pp. 462–497.

[22] E. Rahm and P. A. Bernstein, *A survey of approaches to automatic schema matching*, VLDB Journal, 10 (2001), pp. 334–350.

[23] M. Smiljanic, M. van Keulen, and W. Jonker, *Formalizing the xml schema matching problem as a constraint optimization problem*, in DEXA, K. V. Andersen, J. K. Debenham, and R. Wagner, eds., vol. 3588 of Lecture Notes in Computer Science, Springer, 2005, pp. 333–342.

[24] E. Tsang, *Foundations of Constraint Satisfaction*, Acadimic Press, 1993.

[25] Z. Zhang, H. Che, P. Shi, Y. Sun, and J. Gu, *Formulation schema matching problem for combinatorial optimization problem*, IBIS, 1 (2006), pp. 33–60.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: February 3rd, 2008
Accepted: March 19th, 2008
Extended version received: June 17th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 315–327. http://www.scpe.org

ISSN 1895-1767
© 2008 SCPE

A SOCIOTECHNICAL APPROACH TO KNOWLEDGE MANAGEMENT IN THE ERA OF ENTERPRISE 2.0: THE CASE OF ORGANIK

DIMITRIS BIBIKAS, DIMITRIOS KOURTESIS, IRAKLIS PARASKAKIS[*] ANSGAR BERNARDI, LEO SAUERMANN[†] DIMITRIS APOSTOLOU[‡] GREGORIS MENTZAS[§] AND ANA CRISTINA VASCONCELOS[¶]

Abstract. The increasing need of small knowledge-intensive companies for loosely-coupled collaboration and ad-hoc knowledge sharing has led to a strong requirement for an alternative approach to developing knowledge management systems. This paper proposes a framework for managing organisational knowledge that builds on a socio-technical perspective and considers people as well as technology as two highly interconnected components. We introduce a conceptualised system architecture that merges enterprise social software characteristics from the realm of Enterprise 2.0, and information processing techniques from the domain of Semantic Web technologies. In order to deliver a KM approach that could assist in reducing the socio-technical gap, we suggest deploying such a solution using an integrated sociotechnical implementation methodology.

Key words: knowledge management, socio-technical approach, SMEs, enterprise social software, semantic web technologies, system architecture

1. Introduction.

The majority of today's enterprise knowledge management tools, techniques and methodologies have been developed with large firms in mind [25], and thus adhere to requirements that are inevitably in conflict with the peculiarities of small knowledge-intensive companies [12]. Current Knowledge Management (KM) systems are not only expensive to purchase, but also require the commitment of significant resources to their deployment, maintenance, and daily operation. The amount of effort required for performing activities core to KM systems, such as designing taxonomies, classifying information, and monitoring functionality [33] is disproportionate to the resource capacity of most SMEs. Moreover, typical knowledge management systems place emphasis on predetermined workflows and rigid "information-push" approaches [26] that reflect the philosophy behind working practices in large enterprises.

In contrast, SMEs rely mostly on informal person-to-person communications and people-centric operations [12] that take place in largely ad-hoc and non-standardised ways [33]. By and large, size and structure imply that SMEs have a set of distinctive needs that call for the deployment of a new breed of digital environments for generating, sharing, and refining organisational knowledge. The management of knowledge in idiosyncratic environments such as those of small knowledge-intensive firms can, in effect, significantly benefit from key characteristics of enterprise social software, like lightweight deployment, flexibility and simplicity of use, emergent and self-organising knowledge structures, and collaboration-oriented philosophy.

Nevertheless, in the absence of a knowledge representation scheme to assist in the interpretation of the accumulated information, the evolution of content in a bottom-up fashion may hinder the effectiveness of managing this information and eventually prevent knowledge workers from transforming it into knowledge. To that end, the enhancement of enterprise social software with intelligent information processing capabilities through the use of semantic technologies appears as a rather promising direction. Such a blend would result in considerable improvements to the usability and effectiveness of enterprise social software, and would enable an SME-focused KM system to demonstrate the immediate and profound evidence of benefits needed for knowledge workers to accept it and use it in their every-day activities. The underpinning motivation in this article is that by leveraging enterprise social software applications with semantic information processing and contextual awareness, we can achieve significant benefits in managing content and knowledge, while allowing for informal, people-centred and ad hoc every-day procedures to be employed.

The aim of this paper is to propose an alternative approach to developing organisational knowledge management systems for small knowledge-intensive companies. In contrast to typical approaches, where knowledge

[*] South East European Research Centre (SEERC), 17, Mitropoleos Str., 54624 Thessaloniki, Greece, {dbibikas, dkourtesis, iparaskakis}@seerc.org

[†] German Research Center for Artificial Intelligence—DFKI GmbH, Knowledge Management Department, P.O.Box 2080, 67608 Kaiserslautern, Germany, {ansgar.bernardi, leo.sauermann}@dfki.de

[‡] Department of Informatics, University of Piraeus, 80, Karaoli and Dimitriou Str., Piraeus, GR-18534, Greece, dapost@unipi.gr

[§] Institute of Communication and Computer Systems, National Technical University of Athens, 9, Iroon Polytechniou Str., 15780 Zografou, Athens, Greece, gmentzas@mail.ntua.gr

[¶] The University of Sheffield Regent Court, Department of Information Studies, 211, Portobello Str., S14DP, Sheffield, UK, a.c.vasconcelos@sheffield.ac.uk

management systems require specific processual use, we suggest that focus should be shifted to delivering solutions that can organically adapt to their every-day work practices and problem solving activities without imposing them from outside or above [36]. This approach to enterprise knowledge management aims at the creation of an environment where encouragement of active social interaction between individuals and teams, empowerment of participation, and self-motivated engagement can promote innovation and assist in attaining sustainable competitive advantage. This perspective suggests a combination of the up to date largely disconnected social and technical organisational system views.

The structure of the paper is the following. In the next part of this article, we analyse the main premises of the sociotechnical theory. We investigate this concept, showing the link with the OrganiK knowledge management approach and the attempt for an improved sociotechnical fit. In the third section of this study, we present the OrganiK approach to knowledge management. We discuss the sociotechnical OrganiK knowledge management framework, which comprises of two pillars: a people-centred and a technology-centred knowledge management strands. We outline both of these approaches and illustrate a conceptualised system architecture. In the following part of this article, we illustrate the anticipated OrganiK implementation methodology which is inline with the main foundations of the sociotechnical theory. Next, we outline some implications for both theory and practice. We conclude with current research limitations future investigation directions.

2. Socio-technical Knowledge Management Perspectives. Knowledge management literature has often focused on two seemingly disjoint approaches: people-centred and technology-centred strategies [20, 31]. Nevertheless, it is proposed that overly stressing the importance of either technological or social components of knowledge management can sometimes be misleading and conducive to less effective organisational initiatives, since these two approaches may, in some contexts, be of equal usefulness [3, 42]. Drawing upon the basis of sociotechnical theory we argue that is necessary to equally consider people, technologies and organisational environment (internal as well as external), in order to advance the prospect of successfully deploying knowledge management initiatives [10].

This paper adopts the view, following Lytras and Pouloudi [24], that knowledge management can be seen “as a socio-technical phenomenon where the basic social constructs such as person, team and organisation require support from Information and Communication Technology (ICT) applications” (p. 64). A socio-technical approach to leveraging organisational knowledge considers people and technology as two highly interconnected components of a single system and is applied to the study of the relationships and interactivities between the social and technical structures of an organisation [8]. Furthermore, we consider both technological as well as social structures as contextually and mutually constitutive which are often driven by co-evolutionary incidents to previously unpredicted directions [22, 34].

The tension between the social and technical organisational structures can be difficult to harmonise, however. The mutual constitutive role of people and technology inside organisations leads to a continuous negotiation procedure between these two elements. Technical infrastructures affect organisational behaviour, while social structures of organisations shape technology’s functionality. Orlikowski [34] refers, in this context, to the notion of ‘interpretive flexibility’ of technology to characterise the way in which users constitute and interpret technology through shared understandings and meanings during its design and use. She stresses, nevertheless, that there are limits to the extent interpretive flexibility of technology can be exerted, imposed by the material characteristics of technology itself and by the institutional contexts of its design and development. Hence, there is a co-evolutionary procedure between software systems and the organisational social structures (e.g. individuals and teams) in which each are forced to adapt continually by the modifications of the one another [22].

However, it appears that social requirements are often neglected in the process of designing and implementing organisational knowledge management solutions. Overly emphasising on the technical requirements of such a solution (i. e. hardware and software components) often results in diminished attention for the social requirements of the initiative (i. e. organisational and social issues). Such a practice has led to what has become known as the socio-technical gap [36]. As illustrated in the following graphical representation of this divide (Figure 2.1), the technical sub-system leaves a significant part of the social sub-system virtually unsupported. The sociotechnical gap indicates a weakly supported social sub-system by the technical structures of the organisation.

Sociotechnical theory focuses on the joint optimisation of both technical as well as social structures of the organisation which constitute the total work system [21]. Tools, technical infrastructures, codified knowledge assets necessary to produce certain outputs comprise the technical sub-system of the organisation [16]. On

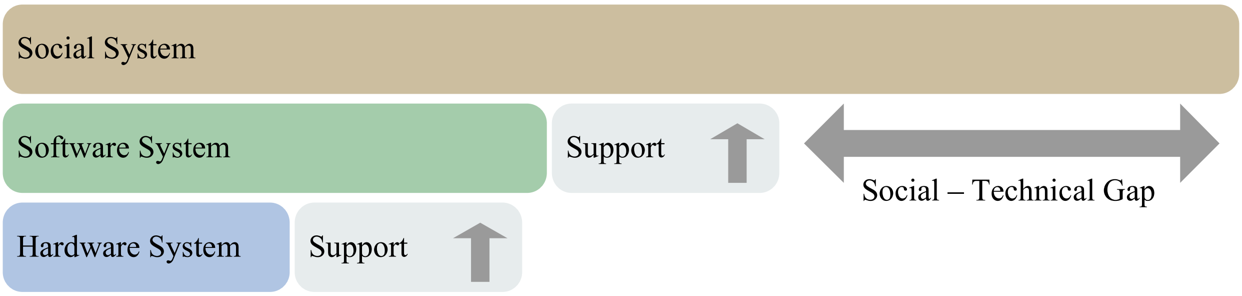


Fig. 2.1. *Socio-technical gap: software and hardware systems provide support for the technical subsystem, while the social subsystem remains virtually unsupported (adapted from [36])*

the other hand, attitudes, beliefs, relationships and results of work arrangements constitute the social subsystem of the organisation [35]. As shown in Figure 2.2, the main premise of sociotechnical studies is the contextual and mutual interdependence of social as well as technical sub-systems of organisations [22]. Post-implementation studies also suggest that often information systems are adapted in use and their organisational role if often reinterpreted and reconstructed through negotiated interaction [7, 11, 13, 40]. Our approach follows the sociotechnical paradigm and studies the relationships and interrelationships between the social and technical parts of the total system [9]. It focused on the interrelated communications which bond the relevant components together and, in accordance with the sociotechnical model it attempts to jointly optimise both elements.

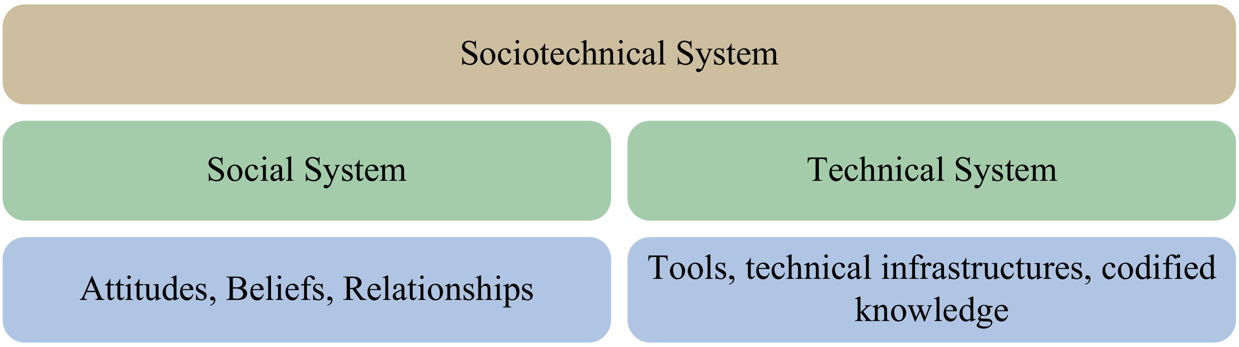

Sociotechnical System
Social System
Technical System
Attitudes, Beliefs, Relationships
Tools, technical infrastructures, codified knowledge

Fig. 2.2. *Sociotechnical theory attempts to jointly optimise both the technical as well as the social structures of the organisation*

We propose an organic perspective to organisational knowledge management system development [36, 10, 29], in which the characteristics of the resulting technical sub-system emerge from a continuous negotiation procedure among the social actors of the organisation and adaptation through user involvement and engagement. This approach attempts to create an iterative dialogic relationship between the social and technical sub-systems that can promote the creation of a collaborative environment for creating, sharing and distilling information in organisational settings.

OrganiK envisions resulting in a knowledge management solution with advanced flexibility and adaptability to current and future needs of the social actors of companies, in which it will be deployed. This knowledge management initiative should result in a technical system with functionalities taking into account the individuals' attitudes, beliefs and social relationships and allowing them to have high level of autonomy in order to engage into every-day problem solving activities. Such a vision is inline with the sociotechnical theory approach which emphasises the link between knowing and action, considering the continuous interplay and mutual constrains of both social and technical organisational sub-systems. OrganiK knowledge management initiative attempts to advance the user involvement and engagement during the system design phase. Furthermore, we conceive the OrganiK knowledge management solution implementation as a procedure of continuous negotiation and inter-play between the organisation's individuals, teams and technical tools. This indicates the creation of an environment in which permanent adaptation and co-evolution of the inseparable nature of systems and people is though to be an important challenge in order to approach an optimsed fit between these two elements. As shown in Figure 2.3 the integrated sociotechnical approach of OrganiK envisions providing enhanced support for the social structures of the organisation and regards implementation and deployment as an ongoing procedure and not as an individual and isolated task.

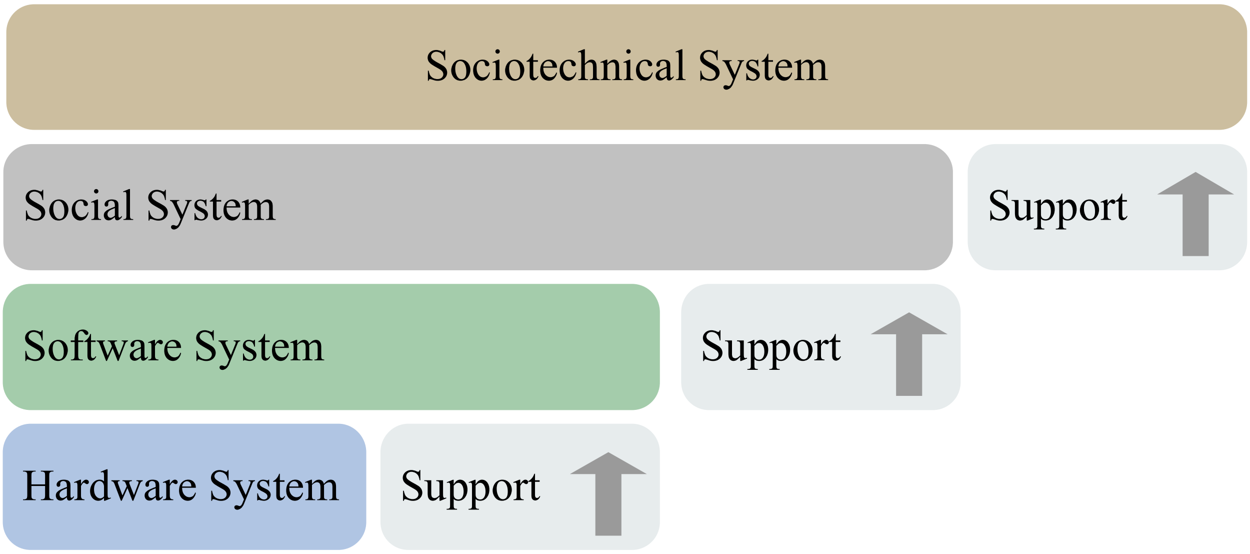


Fig. 2.3. *OrganiK's sociotechnical approach attempts to support both the technical as well as the social structures of the organisation*

3. The OrganiK Approach to Knowledge Management: Towards a Socio-technical fit. An integrated socio-technical knowledge management perspective is a prerequisite in attempting to reduce the divide between the technical and social organisational sub-systems. Therefore, we propose a socially-driven perspective to organisational knowledge management [30], in which the characteristics of the resulting technical sub-system emerge from processes of negotiation among the social actors of the organisation and adaptation through user involvement and engagement. This approach attempts to create an iterative relationship between the social and technical sub-systems and aims at the harmonisation of people and technology inside organisational settings. The vision of the proposed approach is to enable knowledge workers in small knowledge-intensive companies to effectively manage organisational knowledge with the support of an organic knowledge management framework. The major components of the proposed knowledge management framework are the following:

- A people-centred knowledge management conceptualisation, focusing on social processes and work practices of the organisational structures (i. e. individual, team, business units). Situated innovation practices, utilisation of social networks and enhancement of organisational adaptation capabilities comprise fundamental components of this socially-focused approach.
- A technology-centred knowledge management conceptualisation, focusing on the integration of enterprise social software applications (wikis, blogs, collaborative bookmarking tools and search engines) with semantic technologies (ontology-based annotation, semantic text analysis, logic-based reasoning).

Figure 3.1 illustrates the core components of the OrganiK knowledge management framework.

3.1. OrganiK's people-centred knowledge management approach. The OrganiK approach stems from the characteristics and "peculiarities" [12] of knowledge intensive SMEs. The knowledge management literature has often emphasised the lack of uptake of formal knowledge management initiatives in SMEs [28, 43, 33]. However, we propose that there are specific characteristics inherent to SMEs which lead to implicit practices that, although in some ways different to more formal initiatives in larger organisations, can nevertheless, be related to the management of knowledge.

qIt has long been proposed [19, 32] that the size of a company is often correlated with particular structural configurations and patterns and practices of organisational behaviour, namely, the predominance of flatter structures and of task orientation. Emergent and crafted strategies tend to predominate over planned strategies [32], in companies that tend to be more "constrained by resource scarcity" [43] (p. 47) than larger counterparts and therefore may have to adapt faster to survive. Aspects related to sources of power and authority in SMEs

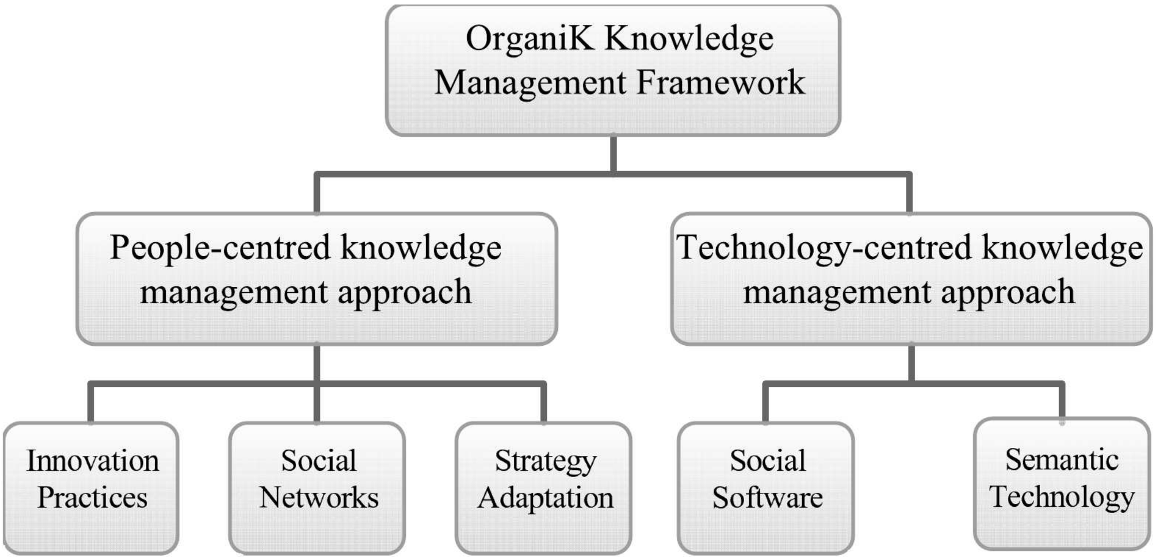

OrganiK Knowledge Management Framework
People-centred knowledge management approach
Technology-centred knowledge management approach
Innovation Practices
Social Networks
Strategy Adaptation
Social Software
Semantic Technology

Fig. 3.1. *The proposed OrganiK knowledge management framework*

remain controversial. Authors such as Handy [19] have in seminal studies emphasised the strength of power cultures in small organisations, centred around the figure(s) of key individual(s), often the founder(s) of the company. Alvesson [1], on the other hand, adds that in the specific case of knowledge intensive SMEs, there tends to be a shift from managerial approaches, based upon direction, planning and control, to less prescriptive and non managerial approaches, where negotiated, rather than explicit sanction-based management, may predominate.

The characteristics of size, structure, behaviour and practices in SMEs can be related, in turn, to different processes of organisational learning and of managing knowledge, as proposed by Desouza and Awazu [12], who, in a case based study of twenty five North American SMEs, identified a series of commonalities in this respect. These include a strong emphasis on socialisation, as the key vehicle for knowledge sharing, and on the tacit common understanding of situations and issues, rather than a reliance on explicit knowledge repositories and formal processes. This leads to two further correlated aspects: i) a strong awareness of the 'common knowledge' of the firm, i. e., knowledge that is known and shared by all its members, and ii) a faster spread of its knowledge base than would be found on larger companies, based on people centred processes, rather than technology centred processes. It appears, therefore, that the organisational learning and knowledge management practices in SMEs tend to be more congruous with apprenticeship based learning, rather than with formal training, and therefore more amenable to management approaches that are more focused on emergence and self regulation, rather than on planning and control [41].

The much debated lack of uptake of formal knowledge management initiatives in SMEs should then be re-thought in terms of focusing on the specificity of the context of SMEs and examining more closely the informal and implicit practices that characterise their organisational learning practices. Knowledge intensive SMEs are an ideal ground to explore this perspective and alternative practices in knowledge management. On the basis of these premises, the people-centred knowledge management approach of the OrganiK framework takes into consideration: i) innovation practices, ii) communities of practice and social networks, and iii) organisational adaptation activities of small knowledge-intensive companies. The following figure illustrates the OrganiK knowledge management people centred pillar. We will now discuss each of its elements in turn.

3.1.1. Innovation practices. The concept of innovation is implicit in many knowledge management definitions and practices [31]. Innovation is often approached as a result of successful knowledge management initiatives and emphasis is placed on the utilisation of knowledge for an organisation to gain enhanced learning and innovation capabilities [24]. In our approach we view knowledge and innovation management as two interlinked processes through a knowledge innovation process model, proposed by Bibikas et al. [5]. Our research draws upon the work of Amidon [2] and explores the concept of Knowledge Innovation, which is defined as: "...the creation, evolution, exchange and application of new ideas into marketable goods and services, leading to the success of an enterprise, the vitality of a nation's economy and the advancement of society" (p. 7). The

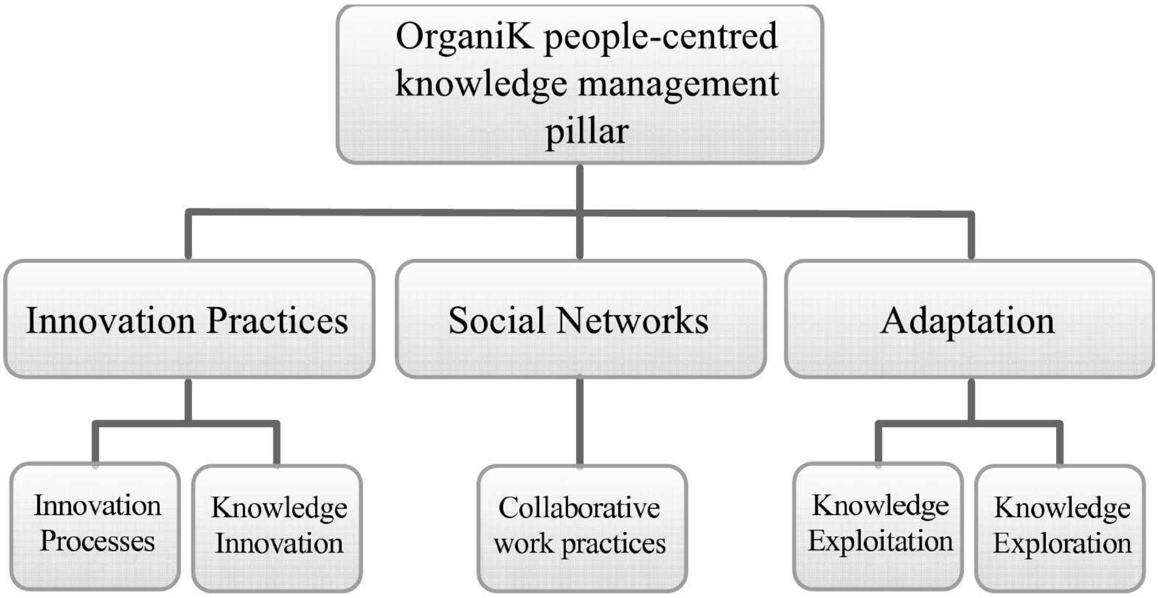


FIG. 3.2. *The proposed OrganiK KM people-centred pillar*

concept of Knowledge Innovation is particularly important to small and medium-sized enterprises (SMEs) which increasingly need to develop their innovation capabilities. This need derives from potential stronger competitive capacities of larger organisations, enabling them to erode traditional SME niche markets.

3.1.2. Communities of Practice and Social networks. The term communities of practice (CoP) was first conceptualised by Lave and Wenger [23] in order to illustrate forms of social organisation independent from formal organisational structures and procedures, binding its members based on similar interests and problem-solving focused activities. Communities of practice are voluntary and emergent groups of people, whose management is based upon self-regulation and a tacit understanding of common interests and shared practices, largely led by mutual trust [14]. In this context, knowledge can be continuously shared and negotiated among social actors, members of these networks [37]. In the OrganiK framework communities of practice and social networks are enabled in a manner which includes more than internal organisational structures (e.g. employees, shareholders, business units, etc), but, rather, integrates elements from the outer environment, such as customers, suppliers, partners and even competitors. CoPs and social networks are of particular importance to the viability of SMEs, since small knowledge-intensive companies usually operate utilising *ad-hoc* and largely social day-to-day collaborative work practices both inside their organisational structures and in their outer business environment.

3.1.3. Organisational adaptation. Typically, organisations manage their cumulative knowledge through two largely defined strategies: knowledge exploitation and knowledge exploration [27]. These perspectives represent two discrete approaches on managing organisational knowledge. Knowledge exploitation entails organisational learning practices which optimise existing processes and improve pre-existing know-how. On the contrary, knowledge exploration consists of organisational learning practices that create new knowledge for the development of novel products, services and processes. However, organisational adaptation requires a balanced adoption of both exploration and exploitation strategies to be successful [27]. Organisational adaptation is of particular importance to SMEs, since their core competitive advantage in relation to larger and globalised firms is their potential rapid responsiveness and quick market adaptation. Boisot [6] suggests that the management of core competences, key to the achievement of competitive advantage, requires the ability to deal with a complex regime that relies on organisations possessing greater and enhanced information processing capabilities than those organisations that do not possess them. We suggest that the management of core competences is based upon the development of adaptive strategies involving the balance between exploration and exploitation for knowledge.

The OrganiK approach aims therefore to support the interplay between active social networks, knowledge innovation processes and organisational adaptation in dynamic knowledge intensive SME contexts, as key elements for competitiveness, through its conceptual framework and the flexibility brought by the integration of enterprise social software applications with semantic technologies.

3.2. OrganiK's technology-centred knowledge management approach. The technology-centred knowledge management approach of the OrganiK framework largely envisions an integration of elements from the domains of Enterprise 2.0 and Semantic Web technologies. We argue that the use of a new breed of emerging collaborative environments in small knowledge intensive organisations can facilitate knowledge work [36, 30, 29]. These new digital environments for generating, sharing and refining knowledge are often popular on the Internet, where they are collectively labelled as "Web 2.0" technologies. Lately, the emerging technologies supporting Web 2.0 applications are entering enterprise bounded environments for creating and sharing organisational knowledge. McAfee [29] introduced the term "Enterprise 2.0" in order to define the employment of social software practices inside organisational settings for information and knowledge management [29].

Although the use of Web 2.0 technologies in business premises can be viewed from varying perspectives and can be referred to employing different names (i. e. social software, social computing, enterprise Web 2.0, Enterprise 2.0, etc), their core operations can be summarised in the following, known as the SLATES framework [29]:

- *Search*, to provide mechanisms for discovering information.
- *Links*, to provide guidance to knowledge workers to discover and later evaluate the needed knowledge while ensuring emergent structure to online content.
- *Authoring*, to enable knowledge workers to widely share their know-how.
- *Tags*, to present an alternative navigational experience exploiting unhierarchical categorisation of content.
- *Extensions*, to exploit collaborative intelligence by suggesting contextually relevant recommendations to knowledge workers.
- *Signals*, to automatically alert knowledge workers for newly available and relevant content.

From a technological point of view the abovementioned SLATES framework is hardly new, since these technologies existed almost since the beginning of the Internet. However, not only are they becoming more and more easy to use, they also convey a novel perspective concerning the process of managing knowledge in organisations. Namely, unlike current knowledge management technologies, where particular tools usually predefine their employment (i. e. presenting certain business rules and somehow inflexible processual requirements), enterprise social software is seemingly abstracted from its practical use. This indicates that the tools are not defining their utilisation in a strict and deterministic manner, while their deployment can be eventually emergent according to adapting needs, ideas, organisational policies etc. As a result, enterprise social software appears to be able to continuously adapt to its environment, a distinctive characteristic of successful enterprise systems [36]. Also, while current enterprise knowledge management software places emphasis on procedural tasks and routine information in a structured manner with specified up front roles, Enterprise 2.0 technologies lets structure emerge, rather than imposing it. In enterprise social software, communication and knowledge sharing structure are to a very large extent self-emerged and organic. Hence, Patrick and Dotsika [36] argue that social software presents enhanced adaptive capabilities with regard to its environment, contrary to the case in which the environment is required to adapt to the functionalities of the software.

Our aim is to provide knowledge workers with a collaborative workspace that comprises a set of integrated Web 2.0 applications, augmented with natural language processing and semantic information integration capabilities. This approach presents two significant benefits. First, the formality of semantics can decrease information ambiguity and increase data interoperability. Information silos across data and applications should communicate with one-another with compatible knowledge models. Second, semantics offer machine-processable characteristics to content, thus making possible knowledge sharing and utilisation activities by means of intelligent software tools [36].

We consider formal knowledge modeling approaches complementary to the dynamic and emergent nature of social software tools. Thus, in our knowledge management technological strand we attempt to merge the formality of semantic technologies with the bottom-up and non-standardised characteristics of enterprise social software.

The use of semantic technologies in the envisaged solution consists of the following key functionalities:

- Semantic knowledge representation: representing knowledge in a formal, machine understandable manner.

- Semantic resource annotation: annotating knowledge artefacts and other resources by reference to concepts defined in an ontological model.
- Semantic inference: performing automated logic-based reasoning to infer new, implicit knowledge based on what has been already asserted in an explicit manner.
- Semantic search and discovery: using ontological terms to describe a search query and rely on logic-based reasoning to derive the matching results.

Each of the aforementioned functions corresponds to one or more of the components in the SLATES enterprise social software framework discussed previously, and, as presented in Figure 3.3, it envisions enhancing enterprise social software basic characteristics.

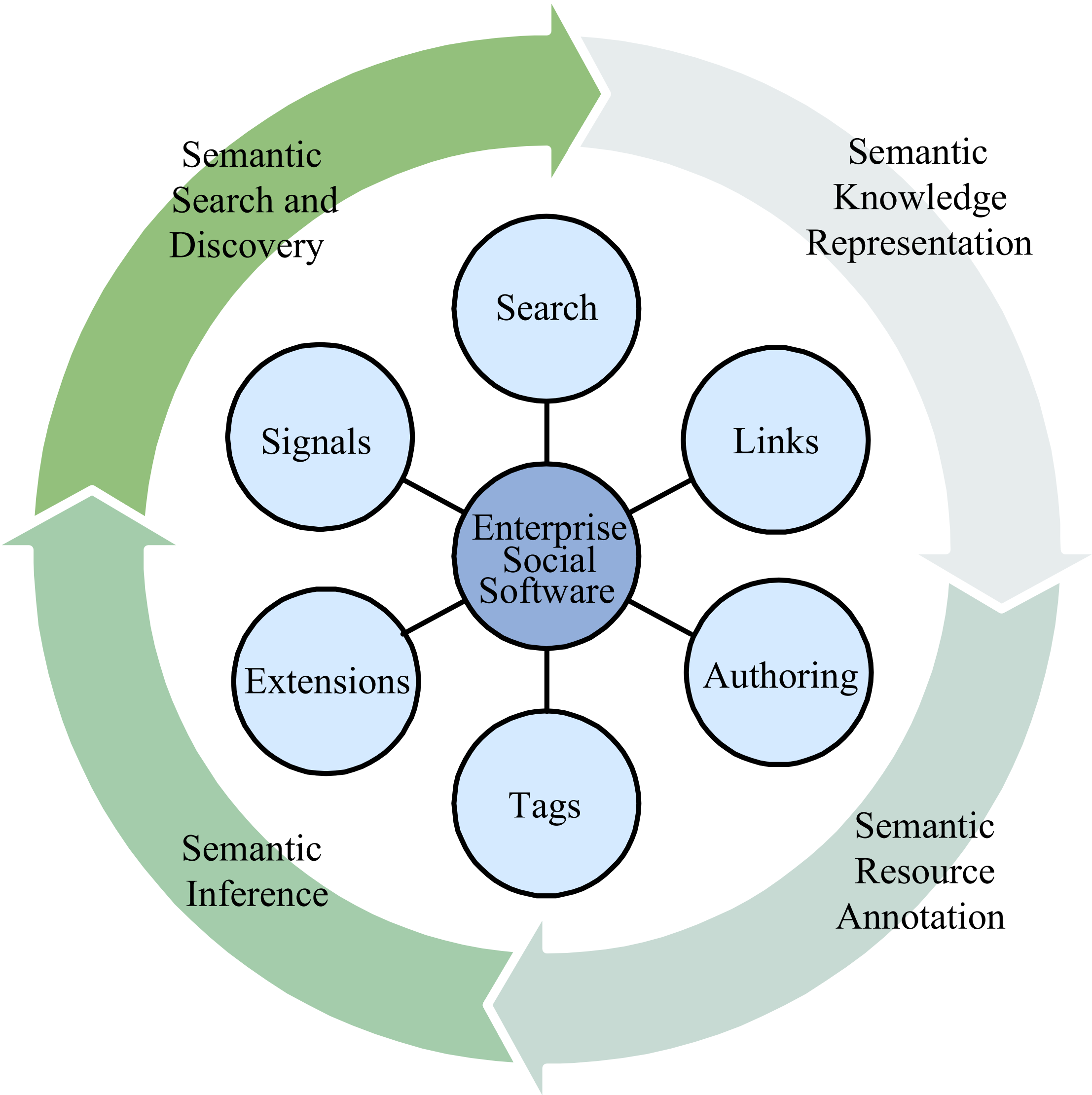

Semantic Search and Discovery
Semantic Knowledge Representation
Semantic Resource Annotation
Semantic Inference
Search
Links
Authoring
Tags
Extensions
Signals
Enterprise Social Software

Fig. 3.3. *Integrating components of the SLATES framework with machine processable semantics*

3.3. Conceptualised Architecture. In this Section we give an overview of the anticipated OrganiK technical architecture. The architecture consists of components that function on different layers, providing the features mentioned in the earlier section. A conceptualisation of the proposed architecture is illustrated in Figure 3.4. The part visible to the end user is represented in the *Client Interface Layer*. It offers a collaborative workspace to knowledge workers and comprises a wiki, a blog, a social bookmarking tool and a search interface.

Each of the client interfaces corresponds to a server-side component in the next layer of the architecture; the *Component Interface Layer*. The server-side building blocks that comprise the *Business Logic Layer* are a recommender system, a semantic text analyser, a collaborative filtering engine and a full-text indexer. Each of the component interfaces are envisioned to access multiple of the services in the business logic layer, yet hiding their complexity from users. The *Metadata Layer* refers to repositories used for the persistence of syntactic and semantic metadata supporting the functionality of all server-side components, while the *Datasources and Back-Office Integration Layer* refers to business information systems and any form of resource container that an enterprise may depend on for its daily operations.

The functionality of the core components in the proposed architecture is envisaged as follows:

- The *Wiki Component* is a web-based authoring tool allowing knowledge workers to collaboratively create, edit, and share knowledge artefacts such as documents, diagrams, etc. The traditional wiki metaphor is extended by the possibility to bind a wiki article to a knowledge artefact, making the wiki page *represent* the knowledge artefact.
- The *Blog Component* provides a simple content management tool enabling knowledge workers to build and maintain open project monitoring diaries, complete with links to relevant resources and user commentary.
- The *Social Bookmarking Component* enables knowledge workers to organise and annotate resources relevant to their activities (e.g. intranet documents, web resources, wiki entries, blog posts, etc) and share them with their co-workers.
- The *Semantic Search Component* supports browsing, filtering, searching, retrieving and displaying knowledge resources leveraging fulltext indexing, semantic annotation indexing, and logic-based inferencing.
- The *Recommender System* focuses on the suggestion of tags and classifications for content added to the system (e.g. wiki entries, bookmarked documents and websites, blog posts and comments, etc.), and the suggestion of information items relevant to the search query or feed subscription of a user.
- The *Semantic Text Analyser* employs linguistic and statistical processing functions on the textual content of knowledge artefacts added to the system, in order to perform named entity recognition and term classification. The objective is to identify concepts of interest and establish relationships among resources that can be subsequently used by the Recommender System for suggesting tags and classifications with respect to a taxonomy/ontology. The metadata created by the Semantic Text Analyser is indexed together with the document in the Metadata Layer.
- The *Collaborative Filtering Engine* enables individual knowledge workers to benefit from the collective experience built within groups of peers. Annotations are envisaged to be created by different users, thus generating an emerging *folksonomy*. This component analyses the subjective views that are explicitly or implicitly expressed by other knowledge workers and generates a model of metadata terms and their relations to users and documents. These can assist in the selection and recommendation of resources, as well as influence the ranking of search results.
- The *Full Text Indexer* is an indispensable component of the architecture's *Business Logic* layer and complements the content retrieval techniques proposed above. Content edited by users is expected to become indexed. It is also envisioned to connect multiple back-office data sources by partially indexing existing data sources and applications for enhanced subsequent retrieval.

Additionally to the presented components, we expect requirements for modifications and changes in this architecture which are bound to come during the design and development of the system. However, the above-mentioned core elements have been known to be needed in order to support the socio-technical implementation methodology we follow. Groza et al. [17] found similar system requirements trough scenarios and end-user interviews during the related NEPOMUK research project.

Components involved in the indexing and metadata storage functions are assembled in a pipe architecture, passing the results of one element as input for the next. IBM's Unstructured Information Management Architecture (UIMA) architecture [18] comprises a role model and good basis for the interaction between these modules. A challenge concerning the technical architecture is to find such role models that fit our requirements and reuse existing frameworks to realise the architecture as such (e.g. frameworks on the architectural abstraction level of Java Platform, Enterprise Edition (Java EE), Service-Oriented Architecture (SOA) frameworks, content management frameworks such as Java Specification Requests 170). The same question of reuse also applies for each individual component.

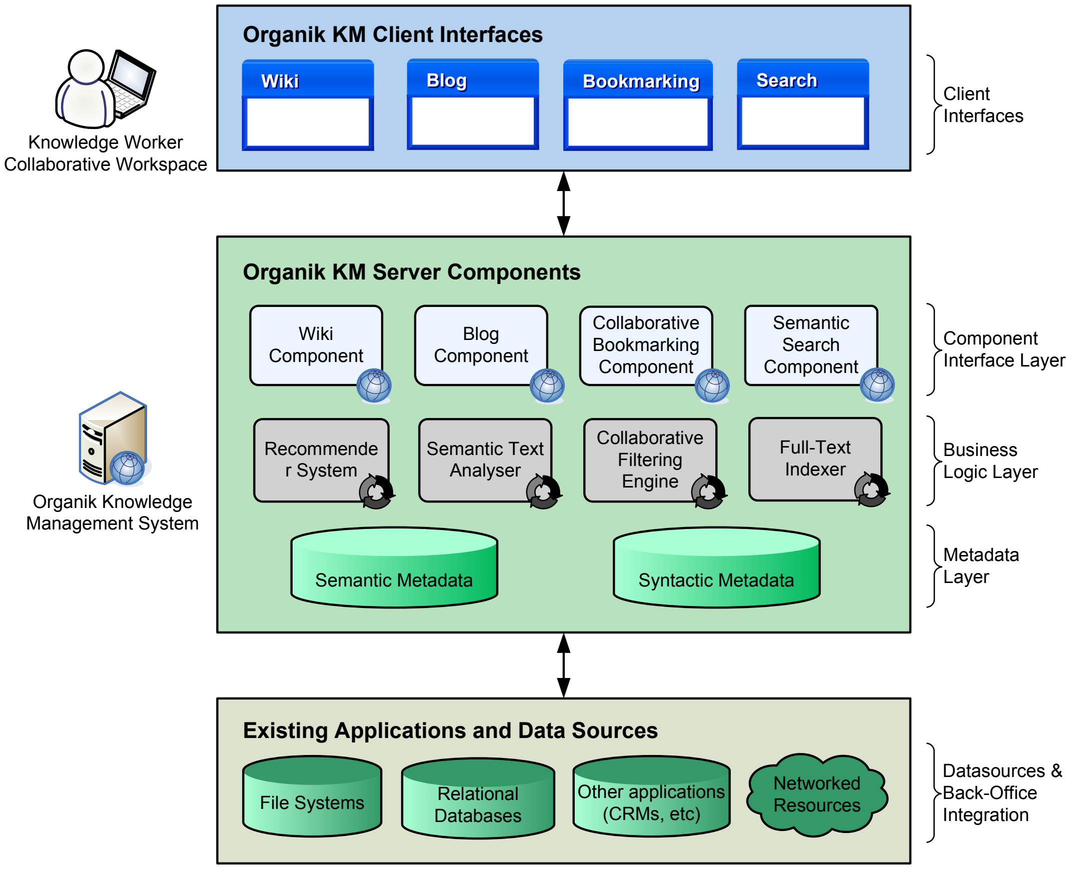

Organik KM Client Interfaces
Wiki
Blog
Bookmarking
Search
Client Interfaces
Knowledge Worker Collaborative Workspace
Organik KM Server Components
Wiki Component
Blog Component
Collaborative Bookmarking Component
Semantic Search Component
Component Interface Layer
Recommende r System
Semantic Text Analyser
Collaborative Filtering Engine
Full-Text Indexer
Business Logic Layer
Semantic Metadata
Syntactic Metadata
Metadata Layer
Organik Knowledge Management System
Existing Applications and Data Sources
File Systems
Relational Databases
Other applications (CRMs, etc)
Networked Resources
Datasources & Back-Office Integration

Fig. 3.4. *Proposed conceptual architecture for semantically-enriched enterprise social software*

Table 3.1
Association among components in SLATES and our proposed architecture

SLATES Framework	Proposed Architecture
Search	Semantic Search
Links	Collaborative Bookmarking
Authoring	Wiki and Blog spaces
Tags	Collaborative Bookmarking, Wiki and Blog spaces
Extensions	Recommender System
Signals	Really Simple Syndication (RSS)

To summarise, the enhancement of enterprise social software tools with machine-processable semantics and their respective processing techniques is expected to yield significant benefits with respect to efficiency of information management, and contribute towards improving the overall user experience of knowledge workers.

Finally, as illustrated in Table 3.1, the proposed OrganiK architecture attempts to integrate enterprise social software's basic characteristics with semantic technologies, since each suggested architectural component corresponds to specific SLATES framework element.

4. Planed sociotechnical Implementation Methodology. The envisioned OrganiK implementation methodology was designed in order to address three significant challenges often found in complex process analysis projects [21]:

- complex technological requirements;
- non-standardised and non-routine knowledge-intensive work processes; and
- considerable social influences in work habits.

Therefore, the expected OrganiK sociotechnical implementation methodology attempts to provide a balanced and holistic analysis of both the social as well as the technical aspects of the investigated processes, in order to implement the final solution. Our approach draws upon the basics of sociotechnical design methodology [15, 39] also taking into consideration its modifications [21]. Our methodology comprises of two parallel studies. The first is focused on the technical subsystem (e.g. infrastructure, software tools, information systems), while the other explores ways to encourage knowledge-worker engagement and involvement. Figure 4.1 below illustrates this integration attempt with regards to the interplay between the social and technical sub-systems.

The OrganiK implementation methodology consists of five phases: Initial Process Scanning, Technical Subsystem Analysis, Social Subsystem Analysis, Interpretation of results, and Solution Design and Implementation. Each phase is discussed below.

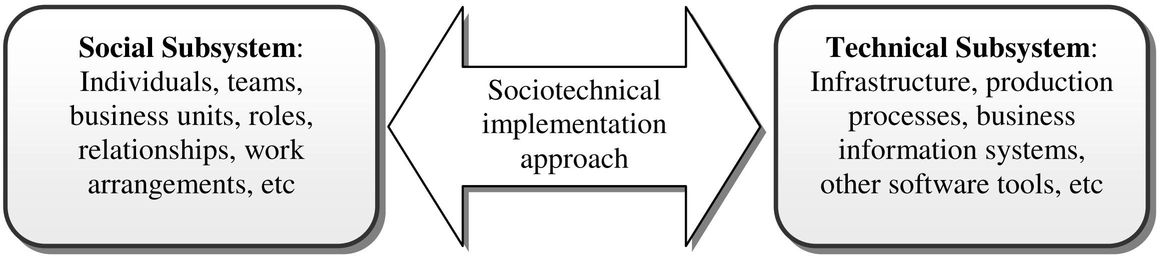

Social Subsystem: Individuals, teams, business units, roles, relationships, work arrangements, etc
Sociotechnical implementation approach
Technical Subsystem: Infrastructure, production processes, business information systems, other software tools, etc

Fig. 4.1. *Integrating social and technical subsystems for the implementation of our solution*

4.1. Phase One: Initial Process Scanning. This first stage of the implementation methodology aims to facilitate a general understanding of the organisation for which the OrganiK solution is implemented for. It is the initial step in order to comprehend the purpose, the process and the environment of the system under review [38]. The scope of that phase is to reveal the main problems on which the analysis should focus [4]. Main work process, general organisational contexts that influence the process (e.g. organisational history, relationships and experiences) are to be investigated in this step. In this phase, the research team is expected to develop boundaries in which the subsequent analysis will take place, as well as a structure and approach for the effort [21]. Once the Initial Process Scanning phase will be complete the analysis will progress to the second phase of the implementation methodology, the Technical Subsystem Analysis.

4.2. Phase Two: Technical Subsystem Analysis. The aim of this phase is to investigate in detail the technical aspects of the total work system [21]. To accomplish such a task we will identify and map the detailed specifications of the main work processes (i.e. their inputs, transformation procedures and final outputs). Furthermore, we will classify the main tools (e.g. business information systems, software tools, intranets, etc) which play a role in the value chain of the organisation and present significant consequences on cost, schedule, quality, or performance. Once the Technical Subsystem Analysis in finished, the results are expected to be jointly evaluated with those of the Social Subsystem Analysis.

4.3. Phase Three: Social Subsystem Analysis. The scope of this phase is to investigate the central elements of the social sub-system of the organisation. The aim is to identify the role of the social structures in the performance of the technical configuration. Social roles, relations and needs of individuals and teams are focal points of such an investigation. Also, social dynamics, organisational design, process context and other non-technical influences are to be explored [21]. The social subsystem analysis phase is expected to take place in parallel with the technical one.

4.4. Phase Four: Analyses Interpretation. The scope of this phase is to blend and integrate the technical and social subsystem analyses. A comprehensible understanding of the holistic sociotechnical work system is the challenge here. Joint optimisation of both subsystems is the prerequisite [21]. The research team is expected to identify all major requirements and integrate both the technical as well as the social aspects for the design of the OrganiK solution.

4.5. Phase Five: Solution Design and Implementation. This last phase of the implementation methodology focuses on the transformation of the abovementioned requirements into technical and social aspects of the OrganiK solution. Details of the technical needs will materialise into concrete software tools, while continuous coaching and support to the social actors will be provided by the research team.

5. Discussion and Future Research. This paper theoretically investigates an approach to developing organisational knowledge management systems for small knowledge-intensive companies. In contrast to other approaches employed in present-day, we suggest that a specific processual use should not be imposed onto knowledge workers, but rather, the provided knowledge management solutions should be able to organically adapt to their every-day work practices and problem solving activities. Despite the fact that the OrganiK research project is still at a rather initial stage, we envisage a system that is utilised and organically incorporated into every-day *ad hoc* and knowledge-intensive SME work practices. Our objective is to realise a KM system with increased social acceptance and a positive impact on reducing the socio-technical gap. In particular, we propose an OrganiK knowledge management framework that adopts a sociotechnical perspective to leveraging organisational knowledge, and considers people and technology as two highly interconnected components. We adopt the intersection of social software and semantic technologies as the technological baseline towards realising this vision, and present a high-level conceptual architecture of the envisaged solution.

Acknowledgment. An earlier version of this paper was presented at the 2nd Workshop on Social Aspects of the Web (SAW 2008), held in conjunction with the 11th International Conference on Business Information Systems (BIS 2008), Innsbruck, Austria, May 2008 and appears in the proceedings for that workshop. We appreciate the helpful comments of the anonymous reviewers, the workshop's participants, committee and chair Dominik Flejter. Research project OrganiK (An organic knowledge management system for small European knowledge-intensive companies) is funded by the European Commission's 7th Framework Programme for Research and Technology Development under Grant Agreement 222225 (Research for the benefit of SMEs).

REFERENCES

[1] M. Alvesson, *Knowledge work and knowledge intensive firms*, Oxford University Press, 2004.

[2] D. M. Amidon, *Innovation Strategy for the Knowledge Economy: The Ken Awakening*, Butterworth-Heinemann, Boston, 1997.

[3] G. D. Bhatt, *Knowledge management in organizations: examining the interaction between technologies, techniques, and people*, Journal of Knowledge Management, 5 (2001), pp. 68–75.

[4] S. Biazzo, *Process mapping techniques and organisational analysis: Lessons from sociotechnical system theory*, Business Process Management Journal, 8 (2002), pp. 42–52.

[5] D. Bibikas, I. Paraskakis, A. Psychogios, and A. Vasconcelos, *Knowledge Ecology in Global Business: Managing Intellectual Capital*, IGI Global, 2008 (in press), ch. An integrated Knowledge Innovation Process Management model: The Case of Skandia.

[6] M. Boisot, *Knowledge Assets, Securing competitive advantage in the information economy*, Oxford University Press, Oxford.

[7] A. D. Brown, *Narrative, politics and legitimacy in an it implementation*, Journal of Management Studies, 35 (1998), pp. 35–58.

[8] A. Cartelli, *Ict and knowledge construction: Towards new features for the socio-technical approach*, The Learning Organization, 14 (2007), pp. 436–449.

[9] E. Coakes, *Knowledge Management in the SocioTechnical World*, Springer-Verlag, London, 2002, ch. Knowledge management: a sociotechnical perspective, pp. 4–14.

[10] ———, *Storing and sharing knowledge: Supporting the management of knowledge made explicit in transnational organisations*, The Learning Organization, 13 (2006), pp. 579–593.

[11] J. Cornford and N. Pollock, *Putting the university online*, SRHE and the Open University Press, 2003.

[12] K. C. Desouza and Y. Awazu, *Knowledge management at smes: five peculiarities*, Journal of Knowledge Management, 10 (2006), pp. 32–43.

[13] B. Doolin, *Power and resistance in the implementation of a medical management information system*, Information Systems Journal, 14 (2004), pp. 343–362.

[14] D. Ellis, R. Oldridge, and A. Vasconcelos, *Community and virtual community*, Annual Review of Information Science and Technology, 38 (2004), pp. 145–186.

[15] F. E. Emery and E. L. Trist, *Sociotechnical Systems: a Sourcebook*, University Associates,, 1978, ch. Analytical model for sociotechnical systems, pp. 120–131.

[16] W. M. Fox, *Sociotechnical system principles and guidelines: past and present*, Journal of Applied Behavioral Science, 31 (1995), pp. 95–105.

[17] T. Groza, S. Handschuh, K. Moeller, G. A. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, and R. Gudjonsdottir, *The nepomuk project - on the way to the social semantic desktop*, in Proceedings of I-Semantics' 07, 2007, pp. 201–211.

[18] T. Götz and O. Suhre, *Design and implementation of the uima common analysis system*, IBM Systems Journal, 43 (2004), pp. 476–489.

[19] C. Handy, *Understanding organizations*, Penguin Business, London, 1983.

[20] M. T. Hansen, N. Nohria, and T. Tierney, *Whatďs your strategy for managing knowledge?*, Harvard Business Review, 77 (1999), pp. 106–116.

[21] C. B. Keating, A. A. Fernandez, D. A. Jacobs, and P. Kauffmann, *A methodology for analysis of complex sociotechnical processes*, Business Process Management Journal, 7 (2001), pp. 33–49.

[22] R. M. Kim and S. M. Kaplan, *Interpreting socio-technical co-evolution: Applying complex adaptive systems to is engagement*, Information Technology & People, 19 (2006), pp. 35–54.

[23] J. Lave and E. C. Wenger, *Situated Learning: Legitimate Peripheral Participation*, Cambridge University Press, 1991.

[24] M. D. Lytras and A. Pouloudi, *Towards the development of a novel taxonomy of knowledge management systems from a learning perspective: an integrated approach to learning and knowledge infrastructures*, Journal of Knowledge Management, 10 (2006), pp. 64–80.

[25] S. Maguire, S. C. L. Koh, and A. Magrys, *The adoption of e-business and knowledge management in smes*, Benchmarking: An International Journal, 14 (2007), pp. 37–58.

[26] Y. Malhotra, *Integrating knowledge management technologies in organizational business processes: getting real time enterprises to deliver real business performance*, Journal of Knowledge Management, 9 (2005), pp. 7–28.

[27] J. March, *Exploration and exploitation in organizational learning*, Organization Science, 2 (1991), p. 71ij87.

[28] H. Matlay, *Organisational learning in small learning organisations: An empirical overview*, Education and Training, 42 (2000), pp. 202–211.

[29] A. P. McAfee, *Enterprise 2.0: The dawn of emergent collaboration*, MIT Sloan Management Review, 47 (2006), pp. 21–28.

[30] G. McKelvie, F. Dotsika, and K. Patrick, *Interactive business development, capturing business knowledge and practice: A case study*, The Learning Organization, 14 (2007), pp. 407–422.

[31] G. Mentzas, D. Apostolou, R. Young, and A. Abecker, *Knowledge networking: a holistic solution for leveraging corporate knowledge*, Journal of Knowledge Management, 5 (2001), pp. 94–106.

[32] H. Mintzberg, *Structures in fives: designing effective organization*, Prentice-Hall, Englewood Cliffs (NJ), 1983.

[33] M. B. Nunes, F. Annansingh, and B. Eaglestone, *Knowledge management issues in knowledge-intensive smes*, Journal of Documentation, 62 (2006), pp. 101–119.

[34] W. Orlikowski, *The duality of technology: rethinking the concept of technology in organizations*, Organization Science, 3 (1992), pp. 398–427.

[35] W. A. Pasmore and J. J. Sherwood, eds., *Designing Effective Organizations: The Sociotechnical Systems Perspective*, Wiley, 1988.

[36] K. Patrick and F. Dotsika, *Knowledge sharing: developing from within*, The Learning Organization, 14 (2007), pp. 395–406.

[37] J. Swan, S. Newell, H. Scarbrough, D., and Hislop, *Knowledge management and innovation: networks and networking*, Journal of Knowledge Management, 3 (1999), p. 262ij275.

[38] J. C. Taylor and D. F. Felten, *Performance by design: Sociotechnical systems in North America*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[39] E. L. Trist, *Perspectives on Organization Design and Behaviour*, Willey, 1982, ch. The sociotechnical perspective, pp. 19–75.

[40] A. Vasconcelos, *The role of professional discourses in the organisational adaptation of information systems*, International Journal of Information Management, 27 (2007), pp. 279–293.

[41] A. C. Vasconcelos, *Dilemmas in knowledge management*, Library Management, 29 (2008), pp. 422–443.

[42] K. Wiig, *People-Focused Knowledge Management. How Effective Decision Making Leads to Corporate Success*, Elsevier Butterworth-Heinemann, Burlington, MA, 2004.

[43] K. Wong and E. Aspinwall, *Characterising knowledge management in the small business environment*, Journal of Knowledge Management, 8 (2004), pp. 44–61.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: January 31st, 2008
Accepted: March 19th, 2008
Extended version received: July 1st, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 329–340. http://www.scpe.org

ISSN 1895-1767
© 2008 SCPE

FROM BUSINESS RULES TO APPLICATION RULES IN RICH INTERNET APPLICATIONS

KAY-UWE SCHMIDT, ROLAND STÜHMER[*] AND LJILJANA STOJANOVIC[†]

Abstract. The increase of digital bandwidth and computing power of personal computers as well as the rise of the Web 2.0 came along with a new web programming paradigm: Rich Internet Applications. On the other hand, powerful server-side business rules engines appeared over the last years and let enterprises describe their business policies declaratively as business rules. This paper addresses the problem of how to combine the business rules approach with the new programming paradigm of Rich Internet Applications. We present a novel approach that reuses business rules for deriving declarative presentation and visualization logic. In this paper we introduce a rule-driven architecture capable of executing rules directly on the client by implementing the Rete algorithm. We propose to use declarative rules as platform independent language describing the application and presentation logic. By means of AJAX we exemplarily show how to use client-side executable rules for adapting the user interface of Rich Internet Applications. We call our approach ARRIA: Adaptive Reactive Rich Internet Applications. In order to show the usability of our approach we explain our approach based on an example taken from the financing sector.

Key words: rich internet application, declarative user interface, rule engine, event condition action rules, AJAX

1. Introduction. Today's business world is characterized by globalization and rapidly changing markets. Thus in recent years business processes do not change yearly but monthly, the product lifecycle has shrunk from months to weeks in some industries and the process execution time has decreased from weeks to minutes as a result of the technological progress over the last few years. On the other side, the life cycle of IT applications stayed constant over time [23]. Business rules already proved their potential of bridging the gap between dynamic business processes and static IT applications. By declaratively describing the policies and practices of an enterprise the business rules approach offers the flexibility needed by modern enterprises.

At the same time with the dawn of the Web 2.0, a new technology for web applications appeared: AJAX [15]. Because of the Web 2.0 and AJAX, Rich Internet Applications (RIAs) emerged from their shadow existence in the World Wide Web. AJAX, in contrast to Adobe Flex (http://www.adobe.com/products/flex), now enables RIAs running in browsers without the need for any additional plug-in. Several Web 2.0 applications use AJAX heavily in order to provide a desktop-like behavior to the user. Now the time seems right for RIAs, because of the broad bandwidth of today's Internet connections, as well as the availability of powerful and cheap personal computers. Besides AJAX, other prominent members of the RIA enabling technologies are: Adobe Flex, Microsoft Silverlight (http://www.microsoft.com/silverlight), OpenLaszlo (http://www.openlaszlo.org), to mention just a few.

Given those two trends observable in today's IT landscape, traditional ways of programming web applications no longer meet the demands of modern Rich Internet applications. So, the strict distinction between declarative business logic and hard coded presentation logic does no longer hold. As web citizens are accustomed to highly responsive Web 2.0 applications like Gmail (http://mail.google.com), web applications based on business rules also have to provide the same responsiveness in order to stay competitive.

In this paper we propose a novel, declarative architecture for RIAs. We coined our proposed system architecture ARRIA which stands for Adaptive Reactive Rich Internet Application. In our architecture all business rules, affecting the UI and not demanding intensive back-end processing, are transferred into a client-readable format at design time. We call these rules in the following application rules. At run-time the application rules are executed directly on the client by a client-side rule engine. That enables a RIA to react straight to user interactions. The event patterns triggering the rules are found by a complex event processing unit. After identifying appropriate events, the application rules, in the form of event condition action (ECA) rules, are executed directly on the client. As a proof-of-concept and in order to evaluate the idea of ARRIAs we prototypically realized a rule-driven RIA using AJAX as client-side technology.

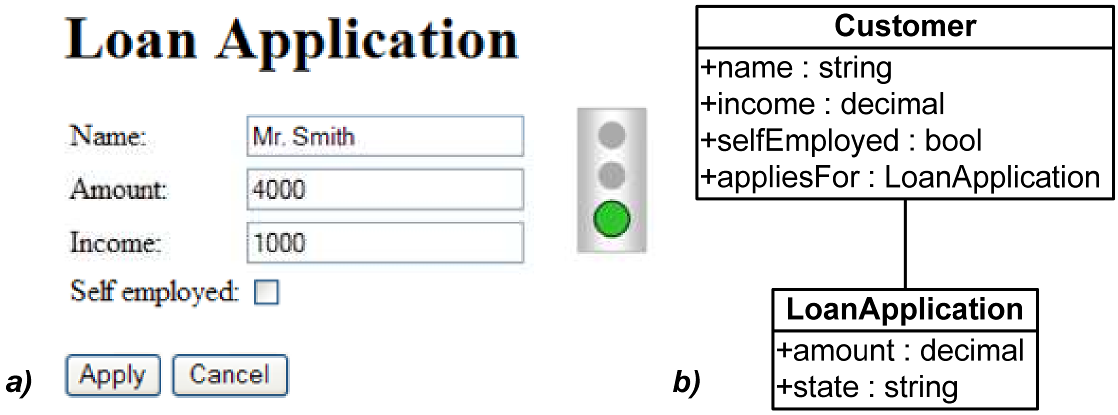
The paper is structured as follows: In Section 2 we present an example in order to motivate our work. The following Section 3 describes the historical development of rule-driven systems. In Section 4 we analyze the semantic and syntactic requirements for a client-side executable rule language. We present in Section 5 our JSON rules approach, an implementation of these requirements. Based on our motivating example we show in

[*] SAP AG, Research, Vincenz-Prießnitz-Straße 1, 76131 Karlsruhe, {Kay-Uwe.Schmidt, Roland.Stuehmer}@sap.com
[†] FZI Forschungszentrum Informatik, Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Stojanovic@fzi.de

Section 6 how to derive application rules from business rules. Additional, in this section we show an exemplary JSON rule for manipulating objects. The architecture of our ARRIA approach is detailed in Section 7. The subsequent Section 8 elaborates on the implementation details. Section 9 gives an overview of related research in the filed of rule-driven RIAs, and, finally, the paper closes with Section 10, conclusions and prospects for future work.

2. Motivating example. For motivating our work we chose an example from the financing sector. The example illustrating our approach is an online application for a loan. The use case is as follows: A person wants to apply for a loan from a bank. S/he visits the web portal of that bank in order to fill in the online loan application. Figure 2.1 shows the form. The web site offers four input possibilities: first, the name of the applicant; second, the amount of the requested loan; third the income of the applicant, and, finally, the kind of employment. The two buttons below the form submit or cancel the loan application.

The IT department of the bank decided to implement the online loan application as RIA in order to take advantage of the advanced visualization techniques. The RIA shall give immediate feedback to the borrower signaling the probability of acceptance. Therefore, a traffic light was additionally introduced on the web page. The lights indicate the status of the application for a loan. A *red* light signals a low or zero probability that the loan will be granted. *Yellow* means that a clerk has to decide whether or not the loan application will be accepted. Finally a *green* light indicates that, based on the input data, the loan will be granted in all probability. The traffic light shall change as the user fills in the online form without explicitly asking the server. That leads to a desktop-like behavior of the web application.


Loan Application

Name: Mr. Smith

Amount: 4000

Income: 1000

Self employed: □

Fig. 2.1. *Motivating example taken from the finance sector*

The business logic of the application for a loan is well understood and written down as business rules, since they are subject to frequent changes. The RIA, and, especially, the manipulation of the traffic light, can reuse and can be built upon these business rules. The rules shown in Example 1 declaratively represent the business logic behind a loan application. For the sake of simplicity we abstract from the amount of the loan. The rules are written in the *IF/THEN* syntax because of its simplicity and its commonness of use.

Example 1 Business Rules.

```
IF C.income {\textgreater}= 1000 AND NOT C.selfEmployed THEN
    L.state = ''accepted''
IF C.income {\textgreater}= 1000 AND C.selfEmployed THEN
    L.state = ''to be checked''
IF C.income {\textless} 1000 THEN L.state = ''rejected''
```

Figure 2.1 b) depicts the UML class diagram of the business objects (BOs). BOs are objects that encapsulate real world data and business behavior associated with the entities that they represent [20]. They are also called objects in a domain model. A domain model represents the set of domain objects and their relationships. The two BOs engaged in our example are *Customer* and *LoanApplication*. They are connected by the relation *appliesFor* which links one customer to one or many loan applications. The attributes of the customer class store customer specific attributes like name, income and employment status, whereas the attributes of the loan class hold loan specific data like the amount or the approval status of the application. A loan application can have the following statuses: *accepted*, *to be checked* or *rejected*.

The business rules depicted in Figure 2.1 a) define the business logic of when to grand a loan application. C is a placeholder for *Customer* objects and L for *LoanApplication* objects. The first rule states that if the borrower's income is grater then or equal to 1000 Euro and s/he is not self employed the loan will be granted in all probability. The second business rule states that if the income is greater then or equal to 1000 and s/he is self employed a clerk has to judge manually whether the loan will be granted or not. If the income is less then 1000 the loan will not be granted at all. In our example, all business rules are atomic. That means they are independent of each other and pairwise disjunct.

3. The evaluation of rule-driven web applications. Legacy rule-driven web applications are based on the web page paradigm as depicted in the left graphic in Figure 3.1. The web page paradigm states that every web page in a series of pages is downloaded separately. User data are collected in forms on the client and are sent to the server by user request. On the server side a production engine processes the input data and executes actions manipulating business objects. Based on the modified business objects a new web page is created and sent back to the client. Business rules in the back-end declaratively describe the business logic of the web application.

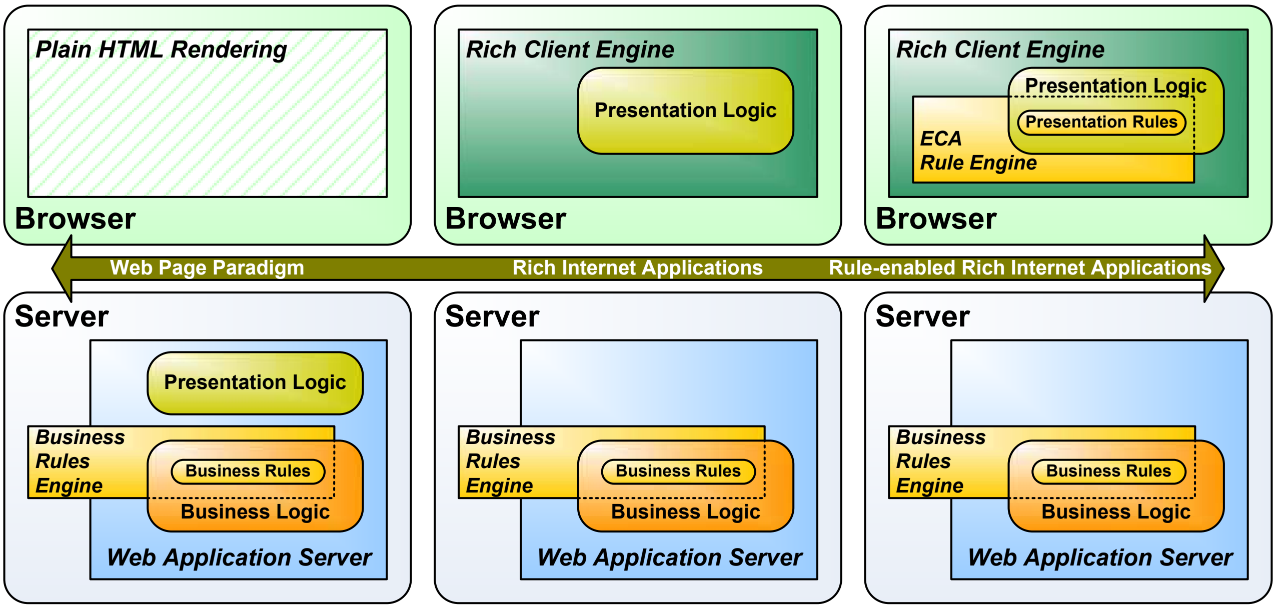


Fig. 3.1. *Evolution of rule-driven web applications*

Rich Internet Applications (RIAs) break the web page paradigm by introducing rich client-side functionality and asynchronous communication facilities. The middlemost graphic in Figure 3.1 depicts the evolution of RIAs from common rule-driven web applications. Up to date browsers provide a rich client engine capable of executing dynamic presentation logics. Together with the business logic, the production system stays on the server side but can be requested asynchronously. That is, business rules can be evaluated without being explicitly triggered by a user request.

Turning RIAs based on server-side rules into client-side rule-driven RIAs, that benefit from the best of the two worlds, is not trivial. Switching from the request/response communication of web applications relying on the web page paradigm to the asynchronous communication of RIAs goes only half way. Although asynchronous communication with the web server allows a RIA to reload only altered data rather then the page as a whole, as well as to pre-load chunks of data that might be good candidates for displaying next, the desired desktop-like responsive behavior is not achieved. This is because business rules and especially business rules concerned with the presentation layer are still evaluated on the server-side. Every user interaction, from pressing a button to hovering the mouse over an artifact on the web site, must be processed on the server in order to let business rules fire appropriate actions as reaction to a user input. Also the advantage of the declarative character of rules is getting lost by only applying the rule paradigm to business logic and not to presentation logic. Presentation logic is also a good candidate for declarative modeling because it remains unchanged even for different platforms.

4. Requirements for a Client-Side Application Rule Language. For managing the proposed client-side rule engine, an appropriate rule language is an indispensable prerequisite. The language must consider requirements specific to Rich Internet Applications.

The semantics of our ECA rule language is constituted by the semantics of the events, conditions and actions by themselves. The semantics of each constituent can be separately defined, for example by reduction to their respective underlying languages. But there is more to it than that. On top of the composed semantics the overall semantics of the language as a whole must be clarified: the relationships between events, conditions and actions.

The so-called coupling modes from early research on ECA rules in the HiPAC project [12] (p.129-143) point out several relationships between events and conditions. However, in all cases a condition is evaluated after an event has occurred. No mode is defined requiring conditions to be fulfilled during the entire occurrence of an event. More recent works, e.g. in [3], suggest a revised semantics for ECA rules. It is stated there that the complete condition of a rule has to be satisfied during the whole detection time of the composite event, i. e. from the beginning of the occurrence of the first constituent event up to the end of the occurrence of its last constituent event. This understanding of ECA rules conforms to the notion of interval-based semantics established for complex events. Interval-based semantics views an event as having a duration, instead of viewing it as an instant at detection time. The duration lasts from the start of the first constituent event to the end of the last constituent event. Therefore, an accompanying condition should span the entire interval of the event duration. The downsides of not using interval-based semantics are pointed out by Galton and Augusto [14] and Berstel [3] for conditions. For events this includes unexpected results from transitivity of multiple sequence operators, and for conditions it includes possible matches with events, violating temporal axioms like matching system context in the future.

Furthermore, the language must expose all user-adjustable features of the event detection, the condition matching and the different kinds of actions. A good complex event detector relies on three things: An easy to use rule language, a rich set of event detection operators and an efficient algorithm to evaluate these operators. Furthermore, the event detection algorithm has to be an active instead of a passive query-based one. We want to stress this a little bit more. Events happen asynchronously and are generally not predicable by nature. Therefore, we insist on a forward-chaining algorithm that pushes actively new events in an appropriate data structure that proactively detects complex events. We are convinced that such a solution outperforms query-based pull strategies for instance proposed by Paschke et al [25].

Conditions are formulae over the state of an application. When a given formula is fulfilled, the system is in a state where the rule author wants some action to be executed. Traditional rule systems only execute condition action rules. The systems are called production systems. Two examples are OPS5 [4] and CLIPS [10]. To evaluate most types of ECA rules, a separate condition matcher is required in addition to the event detection. This can best be observed from the fact that condition action rules lack the triggering event specification, therefore another way must be provided to find and run any applicable rule. Furthermore, any applicable rule should be found at the time when its condition is fully satisfied. This means that changes to the state of the application should immediately be reflected in the activation of rules. No query-driven semantics should be used for rule activation because it would restrict the capacity to act to only certain intervals at which queries are issued. Instead of a query-driven (top-down) approach, a data-driven approach must be employed. A data-driven approach fulfils conditional predicates in a bottom-up way, also called forward-chaining. The advantage of forward-chaining evaluation is that for each change of state affecting a condition, the partial match is saved until it can be further completed to form a complete match in the end. Complete matches are reported immediately when they come into existence.

There are several requirements for the action part of rules. First of all the rule engine should allow for the highest possible flexibility, this means that arbitrary JavaScript actions must be allowed. Apart from the imperative approach using JavaScript, a declarative approach should be supported as it is offered by traditional production systems like OPS5 or CLIPS. In these systems the actions can alter the system state by only specifying modifications to objects. Such modifications include adding and deleting objects, as well as modification of business objects. As a third type of rule action it might be useful to explicitly feed a new event back into the system. Other rules would be able to react to such an event, just like an event from any of the other event sources.

Also the non-functional requirement of user-friendliness targets several aspects. First of all the language should be extensible. This includes permitting the future use of JavaScript features which are not known today. Also, this includes the possibility of adding further operators for the event and condition part. In addition to extensibility, some measures of reusability should be provided. For example, complex event expressions which are repeated in several rules should be made reusable at design-time. The user should have the possibility of creating a set of named event expressions. These predefined expressions can be incorporated into further event expressions of different rules. Methods of reuse should also be provided for condition expressions and possibly actions. For the latter it might be possible to offer a library of predefined actions. User interface patterns [26] might help in finding a meaningful selection of such actions to be provided for the rule author. User-friendliness should also cover the run-time of the rule framework. One important requirement arises when a rule author wants to add and remove rules while the rule engine is running. Both the event detection and the condition matching algorithms must be able to alter their data structures in a coherent manner when rules are added or deleted from detection.

Lastly, on account of browser-friendliness there are also some non-functional requirements for a rule language. As far as the possible acceptance of a new rule language goes, it can be very important that the language closely fits the environment in which it is to be used. To accomplish this, the language should be lightweight, easy to deploy in a RIA and AJAX environment and should honor JavaScript programming practices, where possible.

Luckham writes in his book [22] on event processing, that an event language must be expressive enough, must be notationally simple, semantically precise and must have an efficient pattern matcher. He says this about event languages, but the preceding analysis has shown that Luckham's requirements hold true for the condition part, just as they do for the event part.

5. JSON Rules. We implemented the above analyzed requirements in a rule language named JSON rules. It is a language for defining client-side executable reaction rules. Reaction rules are triples of events, conditions and actions. From a user's point of view the rule language is the interface to programming and adapting ARRIAs. For the rule language the JavaScript-friendly JavaScript Object Notation (JSON) format is chosen. JSON is published as a Request for Comments (RFC) [9]. Like XML it provides a structured representation of data with deep nesting. Unlike XML it is readily usable in JavaScript because JSON syntax is the subset of JavaScript otherwise used to denote objects literals and array literals in the programming language. Although, JSON is JavaScript there is a thin parsing layer involved to provide security from introducing executable code. Other than that, JSON uses a very lean syntax compared to XML. Tags do not need to be named if, for example, they are just used to provide structure like nesting. JSON can be used to maintain nested data; therefore, our rule language can be formulated in JSON as an abstract syntax tree. A similar approach is taken by many modern XML-based languages, like RuleML and its ECA rule standard, Reaction RuleML [25]. Using an abstract syntax tree to transport the language relieves the client-side application of parsing any expressions. Instead the nesting of expressions can be easily determined by descending the supplied tree. Also, no aspects of concrete syntax must be retained when abstract syntax is used.

The complete grammar of our declarative client-side JSON ECA rule language is designed in (extended) Backus-Naur Form (BNF). We designed and tested the rule language grammar with the parser generator tool ANTLR (ANother Tool for Language Recognition, `http://www.antlr.org/`). The grammar describes a so-called rule file. The rule file is the granularity at which rules are transported, e.g. downloaded into the rule framework. A rule file may contain more than one ECA rule in a rule set. Meta data for the rule set are also part of the rule file and a library of reusable event expressions. The language is a specialization of JSON. The syntax of JSON can describe strings, numbers, the Boolean literals *true* and *false* as well as objects and arrays. Objects are enclosed in curly braces. They contain a comma separated list of attributes. An attribute is a string followed by a colon and the value. The value might in turn be any JSON expression. Arrays are enclosed in square brackets, containing a comma separated list of expressions. The proposed language restricts tree-expressions from JSON in way that only certain objects with certain attributes may be used and nested. The language is therefore a subset of JSON. An example JSON rule is given in the next chapter.

6. Deriving application rules from business rules. The starting point for every RIA is the business logic. The business logic declaratively encoded into business rules coarsely defines the presentation logic of the user interface for RIAs. But business rules are usually high-level and are not related to any user interface issues. On the other hand, application rules presenting the presentation logic have to control, on a fine grained level, complex user interfaces. Therefore, the first step in creating application rule sets is the analysis of the business rules and their related business objects. Based on this analysis, the user interface and the presentation logic in the form of declarative application rules can be designed.

The application rule in Example 2 is directly derived from the first business rule of the Example 1. It manipulates all JavaScript *LoanApplication* objects associated with a dedicated *Customer* object, whenever any property of the *Customer* object has changed. A web designer merely has to listen to *PropertyChangedEvents* of the *LoanApplication* object referenced by the *$LoanApp* variable and, if an event has been fired, to adjust the traffic light accordingly. On the other hand, it would also be possible to change the traffic light directly within the rule body by injecting JavaScript code directly into the rule's action part. The printout depicted in Example 2 shows the entire rule set in our case consisting of a single application rule. In line 01 the name of the rule set is defined. From line 02 to 16 a condition action rule is defined. Line 02 states the rule name and line 03 the description of the rule. From line 04 to 13 the condition is formulated. The condition consists of two parts, the first relates to the customer (line 04–08) and the second to the loan application (line 09–13). Line 12 joins all objects of *Customer* meeting the constraints defined in the lines 06–08 with all objects of *LoanApplication* that are not already accepted. In our example the RIA contains only one object of *Customer* and one object of *LoanApplication*. When all constraints are satisfied the action in line 14 is fired.

In line 13 the example JSON rule contains an extra constraint field checking whether `state` is unequal to "accepted". This constraint ensures that the rule is not invoked several times by the execution algorithm. On each change to the rule system runs all rules which have a matched condition. Therefore, a rule fires several times as long as its condition still matches the objects. Since our example rule would always set the loan application to *accepted* regardless of whether this has been done before, the rule would loop endlessly. The solution is to alter the rule in a way so that its condition is invalidated after the rule is run for the first time. Because the rule modifies an attribute which is not part of the condition, we correct this by adding the extra constraint to the rule. The stronger condition ensures that the rule does not match objects which were matched before.

Example 2 JSON Application Rule.

```
01 {"meta": {"ruleSet": ``Loan Application Example"},
02 ``rules": [{"meta": {"rule": ``GrantLoans",
03 ``description": ``Grant loan!"},
04 ``condition": [{"class": ``Customer",
05 ``fields": [
06   {"field": ``income", ``comparator": ``>=", ``literal": 1000},
07   {"field": ``selfEmployed", ``comparator": ``==", ``literal": false},
08   {"field": ``appliesFor", ``vardef": ``$LoanAppID"}]},
09                             {"class": ``LoanApplication",
10 ``vardef": ``$LoanApp",
11 ``fields": [
12   {"field": ``id", ``comparator": ``==", ``variable": ``$LoanAppID"},
13   {"field": ``state", ``comparator": ``!=", ``literal": ``accepted"}]}],
14 ``action": [{"type": ``MODIFY",
15 ``name": ``$LoanApp",
16 ``modify": ``this.state = 'accepted';"}]}]}
```

7. Architecture. First we highlight the design of the CEP engine followed by the design of the rule engine. For the design of an efficient complex event detector several alternative algorithms were proposed in the past. They differ in their detection approach, using either automata [18], Petri-nets [17] or a graph-based approach [8]. They also differ in their effects on the semantics of events they detect, and differ in general versatility.

SnoopIB [1] is chosen from the available approaches as a basis for the event detection in this thesis. Along with that, Snoop's operators are adopted with some extensions and with according extensions of the detection algorithm. A reason for choosing Snoop over the other detection methods is that the graph-based approach of

SNOOP allows the detection of overlapping complex events. This rules out automaton-based event detection as complex events of a given complex type may occur simultaneously. This means several complex incidents of the same type happen at the same time, in an overlapping fashion. Automaton-based algorithms are not capable of detecting more than one instance of the same complex event at the same time. This is an inherent drawback of how automata are used for event detection. As elaborated in Gehani at all [18, 19] automata are constructed from regular expressions specifying event patterns. Transitions model accepted events in a given state. An initial state is created with transitions for initiator events, the initial constituent events. The transitions lead to further states, and so on, up to one or more accepting states, where the complex event is detected. The complex event is then defined as the sequence of transitions which were taken from an initiator to a terminator event. When an automaton must accept overlapping complex events, the following happens. A suitable initiator changes the state of the automaton away from the initial state by using one of the transitions. The automaton will then be in a state which accepts constituent events to continue completion of the first complex event. There might be no transitions accepting initiators for further complex events, until the automaton is reset after completely detecting the first. Although there might be other transitions labeled with the initiator event type, these events will be incorporated in the first complex event as intermediate constituents. Other complex events are only started at the initial state. In summary this means that overlapping complex events are ignored, because once an automaton is in the process of detecting a complex event, it is usually not in its initial state anymore, to start detecting a second complex event at the same time. Algorithms based on Petri nets and on graphs do not share this deficiency. An important drawback of the Petri net-based approach, however, is that Petri nets do not support user-defined selection of tokens when a transition is fired. This means it cannot be predetermined by the user, which constituent events, e.g. tokens, are used when creating a complex event. Therefore, SAMOS [17] does not provide configurable event selection policies in its Petri net-based approach. Coloured Petri nets are introduced in Jensen [21]. They allow tokens to be individually distinguished at the transitions might accomplish event selection based on individual attributes. However, SAMOS uses colours only to model event parameters and to propagate these parameters through the Petri net. This concludes the major reasons for choosing the graph-based approach over automata or Petri nets. Automata cannot detect concurrent complex events and Petri nets do not offer a clear strategy for event selection.

The choice of detection semantics is the next important decision which has to be made on behalf of the event detection. The detection semantics are concerned with whether complex events are represented by an interval or only by a point in time. The preceding analysis for this work showed that a detection-based (point in time) semantics delivers unexpected results for certain operators, e.g. the sequence operator. Snoop revised its semantics towards an interval-based view of events, called SnoopIB [1]. The same holds for other event detection system like Reaction RuleML [25].

According to the requirements we decided to use Rete [13] as forward-chaining discrimination network for the evaluation of the conditions parts of a rule. The Rete algorithm has several similarities with the previously describes detection graph for events. Both are forward-chaining pattern matching algorithms and both must be able to add and remove nodes at runtime, etc. However, there are some important differences. Firstly, it must be noted that they serve different purposes. In terms of semantics of rules [5], the event graph is concerned with transient, temporal data, i. e. events. The Rete network, on the other hand, is concerned with persistent data, representing the system state, i. e. business objects. The two types of data are to be separated in order to avoid making unnecessary events persistent, and thereby imposing a storage burden on an application.

Figure 7.1 depicts the client-side components of the run-time architecture. The server-side components are skipped for the sake of simplicity. The software components of the run-time architecture carry out the application logic encoded in the declarative application rules. The application rules are transferred to the client together with the content data in response to the first initial user request. In the first preprocessing step the CEP Unit responsible for detecting complex events is initialized and, in a second step, the appropriate event handlers are set. As complex events are not issued directly by user interface widgets the CEP Unit has to register for each atomic event contained in complex events.

When the user interacts with the portal, he/she fills in forms, navigates through the site, and goes back, searches for terms and so on. All those interactions trigger events like mouse movements in the appropriate controls. The CEP Unit handles all atomic events to which it has subscribed in advance (step three) by its SnoopIB implementation. Based on the directives of the event detection algebra, it tries to identify complex patterns from the event stream. After detecting a complex event, the associated rules are evaluated by the client-side rule engine. This is step four in Figure 7.1 In step five the condition parts of the rules are evaluated, if there are any, using the Rete algorithm. If the event and condition part of rule is matched during the evaluation phase, it is fired immediately.

The execution of a rule can have manifold actions which are marked as 6a to c. In step 6a a rule manipulates the status of the application. The status of the application is maintained in working memory. In a nutshell, the working memory consists of an arbitrary amount of local object variables. Further a change to the working memory can trigger additional rules that are not explicitly bound to any complex event pattern. These rules are conventional production or condition action rules (CA rules). A rule can also manipulate the user interface directly as depicted in step 6b. By this means, application rules can respond to user interactions immediately without an explicit server request. These rules are the guarantors of a responsive user interface. Any user interface manipulation can issue additional atomic events that might be recognized by the CEP Unit as parts of complex events. New rules can be triggered. So the rule execution in step 6b can trigger additional rules over the event detection mechanism. The last possible action of a rule execution is depicted in step 6c: The invocation of the Asynchronous Communication Controller (ACC). The ACC is responsible for loading new rule sets, for pre-fetching content data as well as for synchronizing with the BO's on the server-side. As a direct byproduct of pre-fetching data and synchronizing with the server, the ACC can alter the user interface.

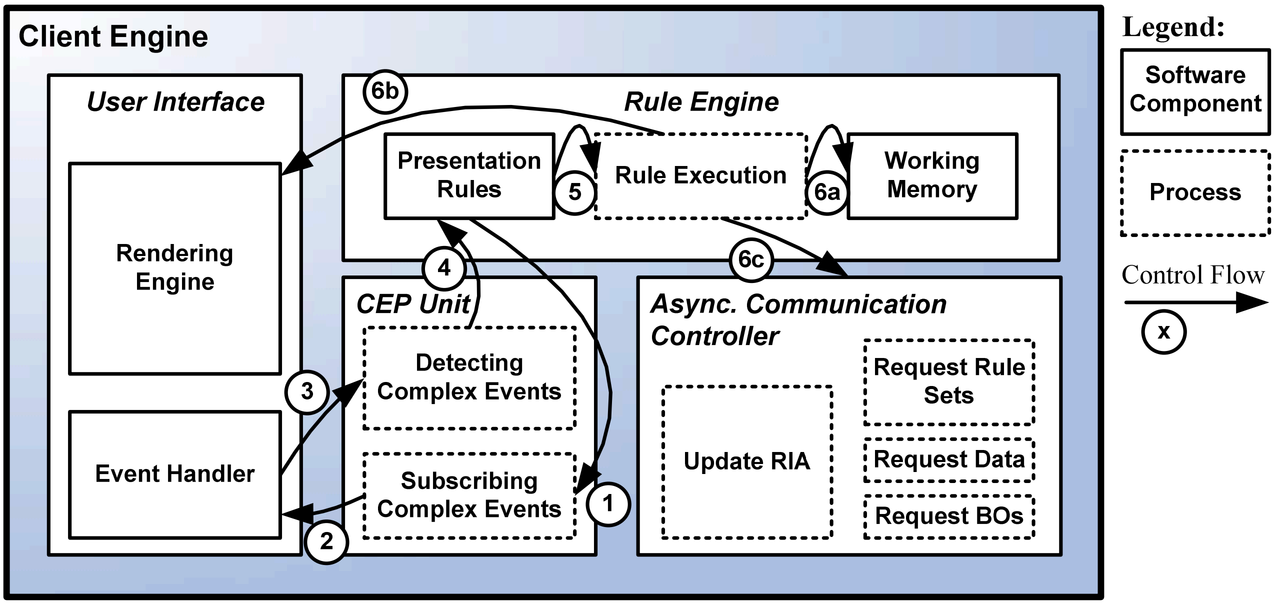


FIG. 7.1. *Run-time architecture*

8. Implementation of the run-time architecture. We implemented our event detection as well as our rule engine in JavaScript using a slightly modified SnoopIB and object-based Rete algorithm. The event graph is a network of nodes which represent event expressions. There are special nodes types for every event type. Incoming edges of a node originate in child nodes which represent sub-expressions. Simple event nodes have no incoming edges. Outgoing edges connect a node to its parent which makes further use of detected events. Detected events are propagated upwards in the network, starting with simple events which are fed into the graph at the simple event nodes. The propagation ends at top nodes which have no further parents. In these nodes events are extracted from the graph and are handed on to some action, which in the process discards the event. Event nodes may have more than one parent. This occurs when an event expression is used in several places of a pattern. The reused expression is then manifested only once in the event graph but outgoing edges are linked to all nodes where the expression is reused. All parent nodes are informed equally of detected events.

We implemented the following event operators in our JSON rule language. The logical operators from Snoop that we implemented are: `Or`, `And`, `Any`, as well as `Not`. Operators `And` and `Or` are binary operators in the sense that they involve two operands. The `Any` operator is a generalized form of the preceding ones. It accepts an arbitrary list of parameters and a parameter m, which specifies the number of events that must be detected to match the Any pattern.

Additional, we implemented Snoop's temporal operators: `Seq`, `A`, `A*`, `P`, `P*`, as well as `Plus`. The operator `Seq` is the sequence of two events in time. Operators `A` and `A*` are ternary operators, detecting occurrences of one event type when they happen within an interval formed by the two other event types. `A*` is a variant which collects all events and occurs only once at the end of the interval with all the collected constituents. `P` and `P*` are ternary operators as well, they also accept two events starting and ending an interval, but the third parameter is a time expression after which the events occur periodically during the given interval. A function may be specified to collect event parameters for each periodic occurrence. `P*` is the cumulative variant which occurs only once, containing all collected constituents. `P` stands for periodic because of its metronome characteristics. `A` stands for aperiodic because the detected constituents occur at irregular times. The `Plus` operator accepts an event type and a time expression. The `Plus` event occurs after the specified event type has occurred and the specified time has passed.

The operators mentioned so far are the complete set from Snoop. Content-based checks are added to them in order to fulfill the requirement for filtering by event parameters. Content-based checks do not provide structure as the previously described operators do. Content based checks filter streams of events, resulting in streams which contain only events matching a constraint check. Such checks are concerned with the parameters of events. The appropriate event operators are called *guard* by David Luckham or *mask* by the authors of Ode. We use the term mask. The event mask is designed as an operator with one event input and a Boolean function to be applied to the input. The value returned from the function decides about whether the input is accepted or discarded. An incoming event is accepted if the function returns `true`. When specifying a mask expression, the function itself may be selected from a set of predefined mask types. Moreover, the event masks in this work are extensible in the way that the function may optionally be an arbitrary user-defined implementation.

The Rete network is constructed from the top downwards, contrary to the event graph. This is because working memory elements (WMEs) enter the Rete network at a single, top node. As with the event graph, equal nodes must be shared. Equality is likewise determined by the function of a node combined with its input, meaning its predecessor nodes. Constructing the Rete network from the rule specification is done as follows. Each class pattern is first converted into a series of consecutive alpha nodes. There are different types of alpha nodes forming sub-classes of Node.Alpha, cf. Figure 8.1. These alpha nodes for example perform checks on the class of an object, the existence of an attribute of an object, or comparisons with the values of attributes, etc. On adding it to the network, each alpha node is linked to its predecessor, checking whether an equal node is already among the successor nodes and sharing it, if so. After the single object checks are completely represented in the network, an alpha memory is added in the end to store the output. To create the successive beta network, joins are gathered from the rule specification. Every free variable occurring in more than one object pattern is invoking a join. Joins are then ordered in pair wise joins by variable and by input memory. Beta nodes are then created with the necessary join predicates and attached to the matching alpha memories. A join predicate or Test is a JavaScript function. It is selected from a hash map of predefined comparator functions which are selected by the comparator specification in each rule. Comparator functions include wrappers for the built-in comparators from JavaScript like $<$, $>$, $<=$, $>=$, $==$, != and like $===$, !== which do not perform type coercions like their two-letter counterparts do. Also the JavaScript special operator *typeof* is available, which allows checks for the types of objects and primitives. Adding more functions to the hash map here provides simple extensibility for the rule framework. The comparator functions are two-parameter functions with Boolean result because they are used as join predicates. The functions are stored in the Test objects in join nodes. A join node has a beta memory as one input and an alpha memory as another. The beta memory supplies tokens which are lists of objects satisfying preceding joins. The alpha memory supplies plain objects (in the form of WMEs) which must match the other objects in the token according to the join predicates. After finishing all joins in the beta memory, a production node is added to the network. Such a node is a beta memory containing finished tokens representing a complete join. Each such token resembles a fully matched pattern and therefore a rule action is triggered from the production node.

9. Related work. Rule-driven Rich Internet Applications seems to be a new and novel approach, as we could not find related work on this topic. Nevertheless, there exists already a reasonable amount of work addressing subtopics of our approach. Carughi at al [6] describe RIAs as reactive systems where the user interface produces events. They use complex event processing in conjunction with server push technologies, but not for triggering application logic formulated in declarative application rules, that can be executed directly on the client. In their work complex events trigger some kind of server-side logic. They also do not address how complex events can be detected on the client-side.

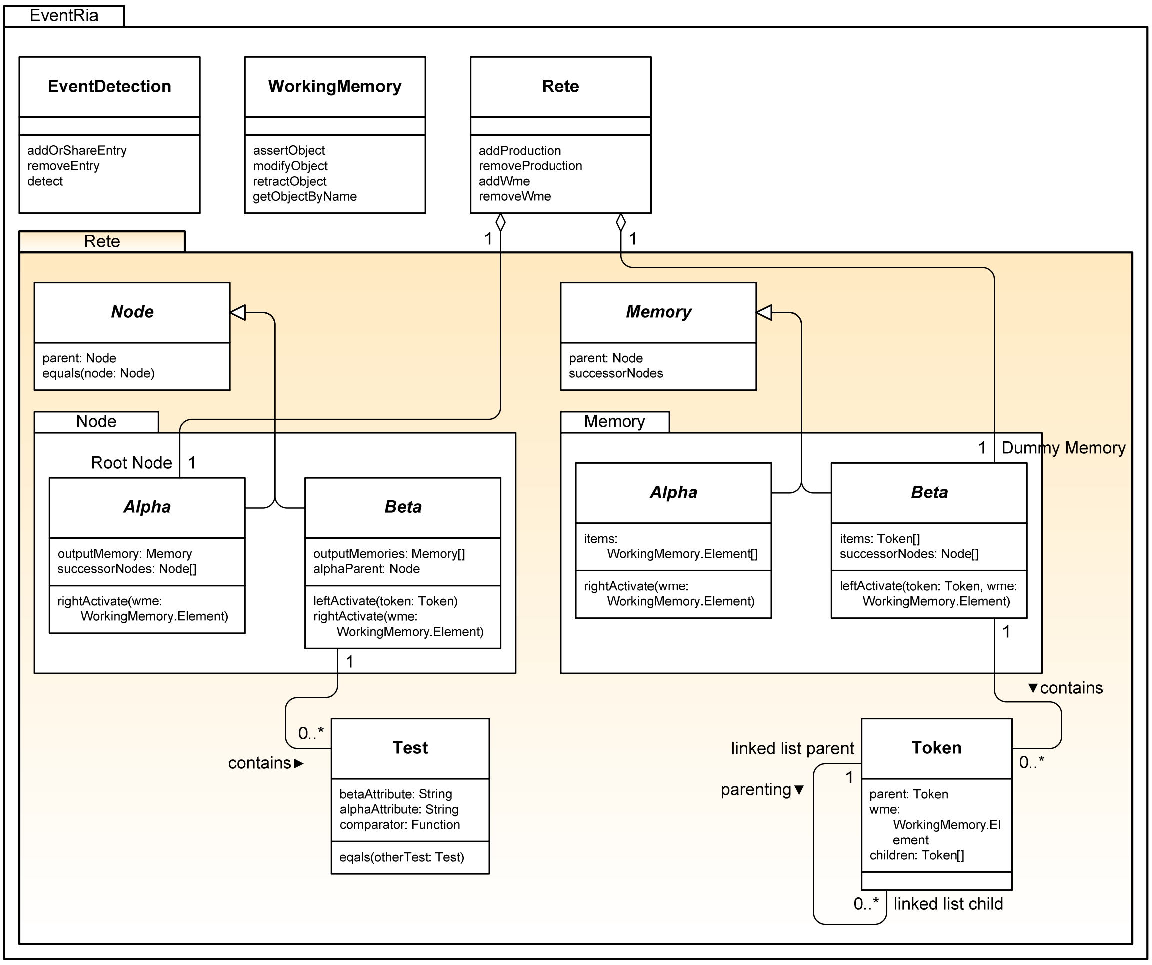


Fig. 8.1. *Rete Network (Class Diagram). This diagram shows the classes comprising the Rete network. The Rete class contains an alpha node in the role of the Root Node. Also, the dummy beta memory is connected to Rete. The rest of the network is reachable through objects of these two classes. Tokens are implemented as a linked list, so token objects are parenting token objects.*

The principles of complex event processing for reactive databases are well understood since the mid-1990s. Chakravarthy et al [8] outline an expressive event specification language for reactive database systems. They also provide algorithms for the detection of composite events and an architecture for an event detector along with its implementation. Our work in the field of complex event processing relies greatly on their work and the work done by Chakravarthy and Mishra [8], Papamarkos et al [24] and Alferes and Tagni [2]. Recently some effort was undertaken to broaden RuleML (`http://www.ruleml.org/`) to a event specification language. As a result Reaction RuleML (`http://ibis.in.tum.de/research/ReactionRuleML/`) [25] incorporates nicely different kinds of production, action, reaction, complex event processing and event logic rules into the native RuleML syntax but fails to support OWL ontologies.

In the web engineering paper of Garrigós et al, [16] AWAC is presented, a prototype CAWE tool for the automatic generation of adaptive web applications based on the A-OOH methodology. The authors define the Personalization Rules Modeling Language (PRML) an ECA language tailored the personalization needs of web applications. Our rule language follows a different approach as it has to deal with complex events on the client-side. PRML does not support complex event processing and is not a general purpose ECA language supporting more then personalization, in contradiction to our JSON rules.

The ECA-Web language suggested by Daniel at al [11] is an enhanced XML-based event condition action language for the specification of active rules, conceived to manage adaptiveness in web applications. Our JSON-Rules are different to that approach as we, as stated in the name, relay on JSON as exchange and execution format. Moreover, we incorporated an event algebra for specifying complex events based on Snoop. Besides that, the whole adaptation approach is quite different as we support real-time adaptation directly on the client compared to the server-side adaptation and rule execution approach of ECA-Web.

10. Conclusions and future work. In this paper we presented a novel approach of using declarative application rules as a new programming model for RIAs. We call this amalgam of event processing, rule engine and RIA: ARRIA – Adaptive Reactive Rich Internet Application. By providing event detection we enable the web designer to define the behavior of the web application based on the order the user issues interaction events in time, that is based on order of his/her actions. The declarative application logic can be easily changed by rewriting the rules. The ECA rules can be executed without additional coding by arbitrary target systems like AJAX, Silverlight or Flex. We developed a light-weight ECA rule language tailored to the needs of RIAs. Furthermore, we implemented an enhanced event detection engine based on the SnoopIB algorithm. For matching the conditions of ECA rules we decided to implement a light-weight version of the Rete algorithm. As a proof of concept we implemented our motivating example using JSON rules. The ARRIA framework consisting of event detection and rule evaluation was implemented in JavaScript. As RIAs are not only AJAX applications we currently implement our framework in Silverlight. Moreover, we will evaluate the performance of the ARRIA framework and we will implement other use cases where our architecture will show its full potential.

REFERENCES

[1] R. Adaikkalavan and S. Chakravarthy, *Snoopib: Interval-based event specification and detection for active databases*, Data Knowl. Eng., 59 (2006), pp. 139–165.

[2] J. J. Alferes and G. E. Tagni, *Implementation of a complex event engine for the web.*, in SCW, IEEE Computer Society, 2006, pp. 65–72.

[3] B. Berstel, *Extending the rete algorithm for event management*, in Proc. Ninth International Symposium on Temporal Representation and Reasoning TIME 2002, Washington, DC, USA, 7–9 July 2002, IEEE Computer Society, pp. 49–51.

[4] L. Brownston, R. Farrell, E. Kant, and N. Martin, *Programming expert systems in OPS5: an introduction to rule-based programming*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1985.

[5] F. Bry and M. Eckert, *Twelve theses on reactive rules for the web*, in EDBT Workshops, 2006, pp. 842–854.

[6] G. T. Carughi, S. Comai, A. Bozzon, and P. Fraternali, *Modeling distributed events in data-intensive rich internet applications.*, in WISE, B. Benatallah, F. Casati, D. Georgakopoulos, C. Bartolini, W. Sadiq, and C. Godart, eds., vol. 4831 of Lecture Notes in Computer Science, Springer, 2007, pp. 593–602.

[7] S. Casteleyn, F. Daniel, P. Dolog, M. Matera, G.-J. Houben, and O. D. Troyer, eds., *Proceedings of the 2nd International Workshop on Adaptation and Evolution in Web Systems Engineering AEWSE'07, Como, Italy, July 19, 2007*, vol. 267 of CEUR Workshop Proceedings, CEUR-WS.org, 2007.

[8] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S. K. Kim, *Composite events for active databases: Semantics, contexts and detection*, in 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile proceedings, J. B. Bocca, M. Jarke, and C. Zaniolo, eds., Los Altos, CA 94022, USA, 1994, Morgan Kaufmann Publishers, pp. 606–617.

[9] D. Crockford, *Rfc4627: Javascript object notation*, tech. report, IETF, 2006.

[10] C. Culbert, G. Riley, and B. Donnell, *Clips reference manual volume 1, basic programming guide, clips version 6.0*, Software Technology Branch, Lyndon B. Johnson Space Center, NASA, (1993).

[11] F. Daniel, M. Matera, A. Morandi, M. Mortari, and G. Pozzi, *Active rules for runtime adaptivity management*, in Casteleyn et al. [7].

[12] U. Dayal, A. P. Buchmann, and D. R. McCarthy, *Rules are objects too: A knowledge model for an active, object-oriented databasesystem*, in Lecture notes in computer science on Advances in object-oriented database systems, New York, NY, USA, 1988, Springer-Verlag New York, Inc., pp. 129–143.

[13] C. L. Forgy, *Rete: a fast algorithm for the many pattern/many object pattern match problem*, Artificial Intelligence, 19 (1982), pp. 17–37.

[14] A. Galton and J. C. Augusto, *Two approaches to event definition*, in DEXA '02: Proceedings of the 13th International Conference on Database and Expert Systems Applications, London, UK, 2002, Springer-Verlag, pp. 547–556.

[15] J. J. Garrett, *Ajax: A new approach to web applications*, `http://www.adaptivepath.com/publications/essays/archives/000385.php` (2005).

[16] I. Garrigós, C. Cruz, and J. Gómez, *A prototype tool for the automatic generation of adaptive websites*, in Casteleyn et al. [7].

[17] S. Gatziu and K. R. Dittrich, *Detecting composite events in active database systems using petrinets*, in Proc. Fourth International Workshop on Active Database Systems Research Issues in Data Engineering, 1994, pp. 2–9.

[18] N. H. Gehani, H. V. Jagadish, and O. Shmueli, *Composite event specification in active databases: Model & implementation*, in VLDB '92: Proceedings of the 18th International Conference on Very Large Data Bases, San Francisco, CA, USA, 1992, Morgan Kaufmann Publishers Inc., pp. 327–338.

[19] ———, *Compose: A system for composite specification and detection*, in Advanced Database Systems, London, UK, 1993, Springer-Verlag, pp. 3–15.

[20] R. Heidasch, *Get ready for the next generation of sap business applications based on the enterprise service-oriented architecture (enterprise soa)*, SAP Professional Journal, 9 (July/August 2007), pp. 103–128.

[21] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods, and Practical Use*, Springer, 1992.

[22] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006.

[23] N. MacDonald, *Strategies for business growth*, in Gartner Symposium ITXPO, 2002.

[24] G. Papamarkos, A. Poulovassilis, and P. T. Wood, *Event-condition-action rule languages for the semantic web*, in SWDB, 2003, pp. 309–327.

[25] A. Paschke, A. Kozlenkov, and H. Boley, *A homogenous reaction rules language for complex event processing*, in International Workshop on Event Drive Architecture for Complex Event Process, 2007.

[26] J. Tidwell, *Designing interfaces*, O'Reilly, 1. ed. ed., 2006.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: December 1st, 2007
Accepted: January 15th, 2008
Extended version received: August 8th, 2008

Scalable Computing: Practice and Experience
Volume 9, Number 4, pp. 341–351. http://www.scpe.org

ISSN 1895-1767
© 2008 SCPE

TRUSTLET, OPEN RESEARCH ON TRUST METRICS

PAOLO MASSA, KASPER SOUREN, MARTINO SALVETTI, AND DANILO TOMASONI*

Abstract. A trust metric is a technique for predicting how much a user of a social network might trust another user. This is especially beneficial in situations where most users are unknown to each other such as online communities. We believe the recent tumultuous evolution of social networking demands for a collective research effort. With this in mind we created Trustlet.org, a platform consisting of a wiki for open research on trust metrics. The goal of Trustlet is to collect and distribute trust network datasets and trust metrics code as Free Software, in order to facilitate the comparison of different trust metrics algorithms and a more coherent progress in this field. At present we made available some social network datasets and code for some trust metrics. In this paper we describe Trustlet and report a first empirical evaluation of different trust metrics on the Advogato social network dataset.

Key words: trust metrics, social network analysis, wiki, advogato, free software, data acquisition, science commons

1. Introduction.

In our current society it is more and more common to interact with strangers, people who are totally unknown to us. This happens for example when receiving an email asking for collaboration or advise from an unknown person, when we rely on reviews written by unknown people on sites such as Amazon.com, and also when browsing random profiles on social networking sites such as Facebook.com or Linkedin.com. Even more surprising is the fact a huge amount of commercial exchanges happen now between strangers, facilitated by platforms such as Ebay.com. In all systems in which is possible to interact with unknown people, it is important to have tools able to suggest which other users can be trustworthy enough for engaging with.

Trust metrics and reputation systems [10] have precisely this goal and become even more important, for instance, in systems where people are connected in the physical world such as carpooling systems or hospitality exchange networks (i. e. couchsurfing.com), in which users accept to have strangers into their car or their house. In fact, in all the previous examples, the system can give users the possibility of expressing a trust statement, an explicit statement stating "I trust this person in this context" (for example as a pleasant guest in a house or as a reliable seller of items) [10] and then use this information in order to predict trustworthiness of users. Trust becomes in this way one of the building block of the society [5].

While research about trust issues spanned disciplines as diverse as economics, psychology, sociology, anthropology and political science for centuries, it is only recently that the widespread availability of modern communication technologies facilitated empirical research on large social networks, since it is now possible to collect real world datasets and analyze them [10]. As a consequence, recently computer scientists and physicists started contributing to this new research field as well [13, 3].

Moreover we all start relying more and more on these social networking sites [4], for friendship, commerce, work, . . . As this field become more and more crucial, in the past few years many trust metrics have been proposed but there is a lack of comparisons and analysis of different trust metrics under the same conditions. As Sierra and Sabater put it in their complete "Review on Computational Trust and Reputation Models" [15]: "Finally, analyzing the models presented in this article we found that there is a complete absence of test-beds and frameworks to evaluate and compare the models under a set of representative and common conditions. This situation is quite confusing, especially for the possible users of these trust and reputation models. It is thus urgent to define a set of test-beds that allow the research community to establish comparisons in a similar way to what happens in other areas (e.g. machine learning)". Our goal is to fulfill this void and for this reason we set up Trustlet [1], a collaborative wiki in which we aim to aggregate researchers interested in trust and reputation and build together a lively test-bed and community for trust metrics evaluation. A related project is the Agent Reputation and Trust (ART) Testbed [6]. However ART is more focused on evaluating different strategies for interactions in societies in which there is competition and the goal is to perform more successfully than other players, in a specific context. Our take with Trustlet is about evaluating performances of trust metrics in their ability to predict how much a user could trust another user, in every context. For this reason, we want also to support off-line evaluation of different trust metrics on social network datasets. The two testbeds are hence complementary.

In this paper we describe Trustlet, the reason behind its creation and its goals, we report the datasets we have collected and released and the trust metrics we have implemented and we present a first empirical evaluation of different trust metrics on the Advogato dataset.

*FBK/rst, Via Sommarive, 14, Povo (TN)—Italy, {massa, souren, salvetti, tomasoni}@fbk.eu

2. Trust Metrics. Trust metrics are a way to measure trust one entity could place in another entity. Let us start with some examples. After a transaction user Alice on Ebay can explicitly express her subjective level of trust in user Bob. We model this as a trust statement from Alice to Bob. Trust statements can be weighted, for example on Advogato [8] a user can certify another user as Master, Journeyer, Apprentice or Observer, based on the perceived level of involvement in the Free Software community. Trust statements are directed and not necessary symmetric: it's possible a user reciprocates with a different trust statement or not at all. By aggregating the trust statements expressed by all the members of the community it is possible to build the entire trust network (for an example, see Figure 2.1). A trust network is hence a directed, weighted graph. In fact trust can be considered as one of the possible social relationships between humans, and trust networks a subclass of social networks [13, 3].

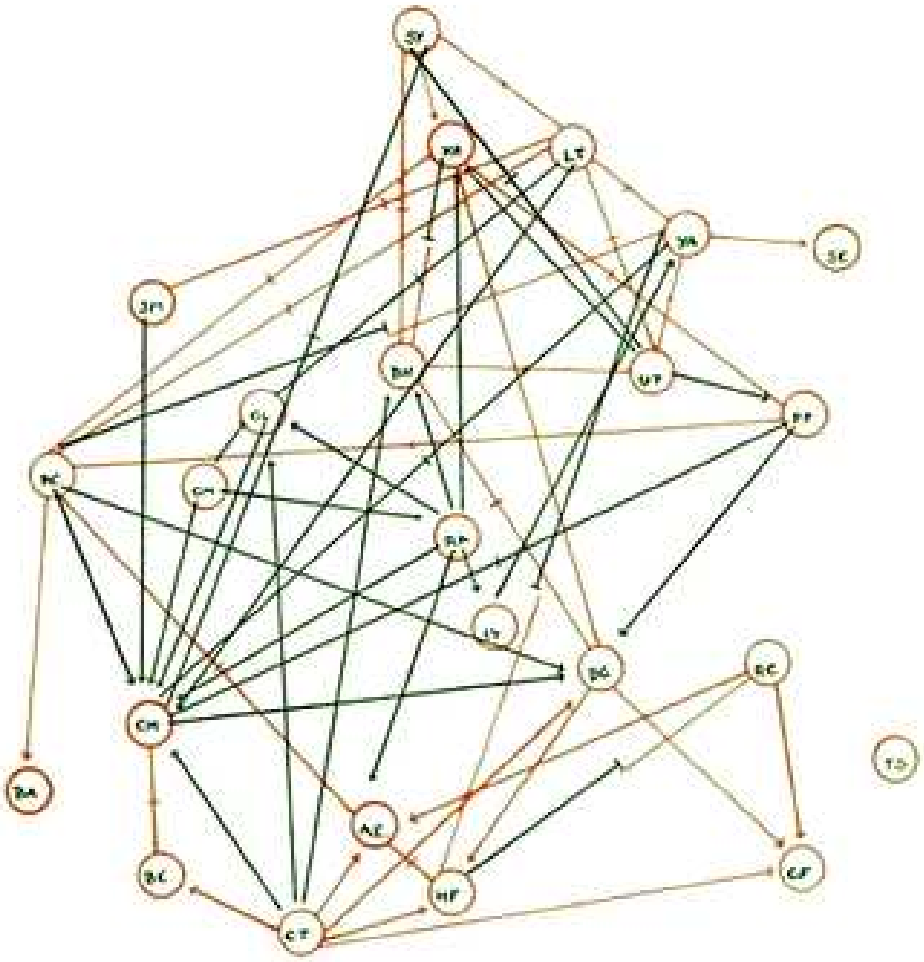


FIG. 2.1. *Structure of a cottage family, hand drawing by Jacob Moreno—From "Who shall survive?" [12]*

Trust metrics are hence tools for predicting the trust a user could have in another user, by analyzing the trust network and assuming that trust can somehow be propagated. One of the assumptions is that people are more likely to trust a friend of a friend than a random stranger [11, 16, 7, 8].

Trust metrics can either be local or global [16, 11]. A global trust metric is a trust metric where predicted trust values for nodes are not personalized.

On the other hand, with local trust metrics, the trust values a user sees for other users depend on her position in the network. In fact, a local trust metric predicts trust scores that are personalized from the point of view of every single user. For example a lo cal trust metric might predict "Alice should trust Carol as 0.9" and "Bob should trust Carol as 0.1", or more formally trust(A,C)=0.9 and trust(B,C)=0.1. Instead for global trust metrics, trust(A,B)=reputation(B) for every user A. This global value is sometimes called reputation [10]. Currently most trust metrics used in web communities are global, mainly because they are simpler to understand for the users and faster to run on central servers since they have to be executed just once for the entire community. However we think that soon users will start asking for systems that take into account their own peculiar points of view and hence local trust metrics, possibly to be run in a decentralized fashion on their own devices.

While research on trust metrics is quite recent, there have been some proposals for trust metrics. We briefly review some of them for later mention in the evaluation presented in Section 4, although our goal is not to provide a complete review of previously proposed trust metrics here.

Ebay web site shows the average of the feedbacks received by a certain user in her profile page. This can be considered as a simple global trust metric, which predicts, as trust of A in B, the average of all the trust statements received by B [11].

In more advanced trust metrics, trust can be extended beyond direct connections. The original Advogato trust metric [8] is global, and uses network flow to let trust flow from a "seed" of 4 users, who are declared trustworthy a priori, towards the rest of the network. The network flow is first calculated on the network of trust statements whose value is Master (highest value) to find who classifies as Master. Then the Journeyer edges are added to this network and the network flow is calculated again to find users who classify as Journeyer. Finally the users with Apprentice status are found by calculating the flow on all but the Observer edges. The untrusted Observer status is given if no trust flow reached a node. By replacing the 4 seed users for an individual user A, Advogato can also be used as a local trust metrics predicting trust from the point of view of A.

The problem of ranking of web pages in the results of a search engine query can be regarded under a trust perspective. A link from page A to page B can be seen as a trust statement from A to B (in this case, the nodes of the trust network are not people but Web pages). This is the intuition behind the algorithm PageRank [2] powering the search engine Google. Trust is propagated with a mechanism resembling a random walk over the trust network.

Moletrust [11] is a local trust metric. Users are ordered based on their distance from the source user, and only trust edges that go from distance n to distance $n + 1$ are regarded. The trust value of users at distance n only depends on the already calculated trust values at distance $n - 1$. The scores that are lower than a specific threshold value are discarded, and the trust score is the average of the incoming trust statements weighted over the trust scores of the nodes at distance $n - 1$. It is possible to control the locality by setting the trust propagation horizon, i.e. the maximum distance to which trust can be propagated.

Golbeck proposed a metric, TidalTrust [7], that is similar to Moletrust. It also works in a breadth first search fashion, but the maximum depth depends on the length of the first path found from the source to the destination. Another local trust metric is Ziegler's AppleSeed [16], based on spreading activation models, a concept from cognitive psychology.

3. Datasets and Trust Metrics Evaluation. Research on trust metrics started a long time ago, but is somehow still in its infancy. The first trust metric could probably be ascribed to the philosopher John Locke who in 1680 wrote: "Probability then being to supply the defect of our knowledge, the grounds of it are these two following: First, the conformity of anything with our own knowledge, observation and experience. Secondly, the testimony of others, vouching their observation and experience. In the testimony of others is to be considered: (1) The number. (2) The integrity. (3) The skill of the witnesses. (4) The design of the author, where it is a testimony out of a book cited. (5) The consistency of the parts and circumstances of the relation. (6) Contrary testimonies" [9]. This quotation can give an idea of how many different models for representing and exploiting trust have been suggested over the centuries. However of course John Locke in 1680 didn't have the technological means for empirically evaluating his "trust metric". Even collecting the required data about social relationships and opinions was very hard in old times. The first contributions in analyzing real social networks can be tracked down to the foundational work of Jacob Moreno [12] (see Figure 2.1) and since then many sociologists, economists and anthropologists have researched on social networks and trust. But the advent of the information age has made it possible to collect, represent, analyze and even build networks way beyond what is possible with pen and paper. Computer scientists and physicists have hence become interested in social networks, now that both huge amounts of data have become available and computing power has advanced considerably [13, 3].

At Trustlet (`http://www.trustlet.org`) we have started a wiki to collect information about research on trust and trust metrics. Our goal is to attract a community of people with interest in trust metrics. The wiki is totally open: anonymous edits are allowed and anybody can register and create an account. We have chosen to use the Creative Commons Attribution license so that work can easily (and legally) be reused elsewhere. Our effort shares the vision of the Science Commons project[1] which tries to remove unnecessary legal and technical barriers to scientific collaboration and innovation and to foster open access to data. We have also started a repository of the software we create for our analysis, written in Python and available as Free Software under the GNU General Public License (GPL) [2] so that other researchers can replicate our experiments and reuse our code.

We believe the lack of generally available datasets is inhibiting scientific progress. It's harder to test a hypothesis if it has been tested on a dataset that is not easily available. The other alternative is testing the hypothesis on synthesized datasets, which are hardly representative of real-world situations. Prior to the proliferation of digital networks data had to be acquired by running face-to-face surveys, which could take years to collect data of a mere couple of hundreds of nodes. The proliferation and popularity of on-line social networks [4] has facilitated acquiring data, and the implementation of standards like XFN and common APIs like OpenSocial opens up new possibilities for research [10]. A more widespread availability and controlled release of datasets would surely benefit research and this is one of the goals behind the creation of Trustlet.

We think it is important that research on trust metrics follows an empirical approach and it should be based on actual real-world data. Our goal with Trustlet is to collect as many datasets as possible in one single place and release them in standard formats under a reasonable license allowing redistribution and, at least, usage in a research context. At present, as part of our e effort with Trustlet, we collected and released datasets derived from `advogato.org`, `people.squeakfoundation.org`, `robots.net`, `kaitiaki.org.nz` and `epinions.com`[3].

We describe in detail the Advogato dataset since our experiments (presented in Section 4) are run on it. Advogato.org is an online community site dedicated to Free Software development, launched in November 1999. It was created by Raph Levien, who also used Advogato as a research testbed for testing his own attack-resistant trust metric, the Advogato trust metric [8]. On Advogato users can certify each other at several levels: Observer, Apprentice, Journeyer or Master. The Advogato trust metric uses this information in order to assign a global certification level to every user. The goal is to be attack-resistant, i. e. to reduce the impact of attackers [8]. Precise rules for giving out trust statements are specified on the Advogato site. Masters are supposed to be principal authors of an "important" Free Software project, excellent programmers who work full time on Free Software, Journeyers contribute significantly, but not necessarily full-time, Apprentices contribute in some way, but are still acquiring the skills needed to make more significant contributions. Observers are users without trust certification, and this is also the default. It is also the level a user certifies another user at to remove a previously expressed trust certification. Notwithstanding the suggestions, users are free to express totally subjective certifications on other users.

For the purpose of this paper we consider these certifications as trust statements [11]. T(A,B) denotes the certification expressed by user A about user B and we map the textual labels Observer, Apprentice, Journeyer and Master in the range [0,1], respectively in the values 0.4, 0.6, 0.8 and 1.0. This choice is arbitrary and considers all the certifications are positive judgments, except for Observer which is used for expressing less-than-sufficient levels. For example, we model the fact raph certified federico as Journeyer as T(raph, federico)=0.8.

The Advogato social network has a peculiarly interesting characteristic: it is almost the only example of a real-world, directed, weighted, large social network. However, besides the leading work of Levien reported in his unfinished PhD thesis [8], we are just aware of another paper using the Advogato dataset which is focused on providing a trust mechanism for mobile devices [14].

There are other web communities using the same software powering Advogato.org and they have the same trust levels and certifications system: `robots.net`, `people.squeakfoundation.org`, `kaitiaki.org.nz`. We collected daily snapshots of all these datasets and made them available on Trustlet but we haven't used them for our analysis in this paper, mainly because they are much smaller than the Advogato dataset. Details about the characteristics of the analyzed Advogato trust network dataset are presented in Section 4.

The other datasets we released on Trustlet are derived from Epinions.com, a website where users can leave reviews about products and maintain a list of users they trust and distrust based on the reviews they wrote [11].

On Trustlet, we released these datasets but our aim is to collectively make it a repository of all the possible datasets useful for research on trust issues. For this reason, we also keep on the Trustlet wiki a list of datasets we are considering for collection and a list of datasets released elsewhere.

[1] Science Commons `http://sciencecommons.org`

[2] GNU General Public License `http://www.gnu.org/licenses/gpl.html`

[3] See `http://www.trustlet.org/wiki/Trustnetworkdatasets`

Moreover, besides aiming at releasing datasets in a coherent format, we also released on Trustlet.org the Python code we wrote for the trust metrics analyzed in Section 4 under a Free Software license so that code can be reused and inspected.

4. Initial Research Outcomes. In the previous sections we highlighted the reasons for creating Trustlet and the way we aim it can develop into a collaborative environment for the research of trust metrics. As a first example of what we envision Trustlet will be able to bring to research on trust metrics, we report our first investigation and empirical findings.

We chose to start studying the Advogato social network because of its almost unique characteristic: trust statements (certifications) are weighted and this makes it a very peculiar dataset for researching trust metrics, in fact, most other networks just exhibit a binary relationship (either trust is present or not) and the evaluation on trust metrics performances is much less insightful.

In this paper we report experiments performed on the Advogato dataset we downloaded from the web site on May 12th 2008. This dataset is available at Trustlet.org, along with datasets downloaded in other days as well. The Advogato dataset under analysis is a directed, weighted graph with 7294 nodes and 52981 trust relations. There are 17489 Master judgments, 21977 for Journeyer, 8817 for Apprentice and 4698 for Observers. The dataset is comprised of 1 large connected component, comprising 70.5% of the nodes; the second largest component contains 7 nodes. The mean in- and out-degree (number of incoming and outgoing edges per user) is 7.26. The mean shortest path length is 3.75. The average cluster coefficient [13] is 0.116. The percentage of trust statements which are reciprocated (when there is a trust statement from A to B, there is also a trust statement from B to A) is 33%.

While a large part of research on social networks focuses on exploring the intrinsic characteristics of the network [13, 6, 3], on Trustlet we are interested in covering an area that received much less attention, analysis of trust metrics. We have compared several trust metrics through leave-one-out, a common technique in machine learning. The process is as follows: one trust edge (e.g. from node A to node B) is taken out of the graph and then the trust metric is used to predict the trust value A should place in B, i. e. the value on the missing edge. We repeat this step for all edges to obtain a prediction graph, in which some edges can contain an undefined trust value (where the trust metric could not predict the value). The real and the predicted values are then compared in order to derive several evaluation measures: the coverage, which is a measure of the edges that were predictable, the fraction of correctly predicted edges, the mean absolute error (MAE) and the root mean squared error (RMSE). Surely there are other ways of evaluating trust metrics: for instance, it can be argued that an important task for trust metrics is to suggest to a user which other still unknown users are more trustworthy, such as suggesting a user worth following on a social bookmarking site such as del.icio.us or on a music community such as Last.fm. In this case the evaluation could just concentrate on the top 10 trustworthy users. But in this first work we considered only leave-one-out as evaluation technique.

4.1. Evaluation of trust metrics on all trust edges. Table 4.1 reports our evaluation results of different trust metrics on the Advogato dataset. It is a computation of different evaluation measures on every edge of the social network. The reported measures are: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), fraction of wrong predictions, and coverage. We now describe the compared trust metrics. As already mentioned we released the code and we plan to implement more trust metrics and release them and run more evaluations.

The compared trust metrics are some trivial ones used as baselines such as Random, which predicts simply a random trust score between the 4 possible ones (1.0, 0.8, 0.6, 0.4), or the metrics starting with “Always” which always return the corresponding value as predicted trust score, for example AlwaysApprentice returns 0.6 for every prediction. Other simple trust metrics are OutA which, in predicting the trust user A could have in user B, simply does the average of the trust statements outgoing from user A, and OutB which averages over the trust statements outgoing from user B. These simple trust metrics are considered in order to understand how much and in which cases complex algorithms are useful.

The other trust metrics were already explained in Section2, here we just report the parameters we used in running them. Ebay refers to the trust metric that, in predicting the trust user A could have in user B, simply does the average of the trust statements incoming in user B, i. e. the average of what all the users think about user B. MoletrustX refers to Moletrust applied with a trust propagation horizon of value X. The values returned by PageRank as predicted trust follow a powerlaw distribution, there are few large PageRank scores and many tiny ones. So we decided to rescaled the results simply by sorting them and linearly mapping them in

Table 4.1
Evaluation of trust metrics on all trust edges

	Fraction wrong predictions	MAE	RMSE	Coverage
Random	0.737	0.223	0.284	1.00
AlwaysMaster	0.670	0.203	0.274	1.00
AlwaysJourneyer	0.585	0.135	0.185	1.00
AlwaysApprentice	0.834	0.233	0.270	1.00
AlwaysObserver	0.911	0.397	0.438	1.00
Ebay	0.350	0.086	0.156	0.98
OutA	0.486	0.106	0.158	0.98
OutB	0.543	0.139	0.205	0.92
Moletrust2	0.366	0.090	0.160	0.80
Moletrust3	0.376	0.091	0.161	0.93
Moletrust4	0.377	0.092	0.161	0.95
PageRank	0.501	0.124	0.191	1.00
AdvogatoLocal	0.550	0.186	0.273	1.00
AdvogatoGlobal	0.595	0.199	0.280	1.00

the range [0.4, 1], after this we rounded the predicted trust scores. Our implementation of Advogato is based on Pymmetry, whose code is released on Trustlet as well. AdvogatoGlobal refers to the Advogato trust metric run considering as seeds the original founders of Advogato community, namely the users "raph", "federico", "miguel" and "alan". This is the version that is running on the Advogato web site for inferring global certifications for all the users. This version is global because it predicts a trust level for user B which is the same for every user. AdvogatoLocal refers to the local version of Advogato trust metric. For example, when predicting the trust user A should place in user B, the trust flow starts from the single seed "user A". This version is local because it produces personalized trust predictions which depends on the current source user and can be different for different users. AdvogatoLocal was run on a subset (8%) of all the edges since the current implementation is very slow. In fact, the leave-one-out technique requires the network be different for every evaluation and it has to be rebuilt from scratch for every single trust edge prediction making the entire process very slow.

Since some trust metrics such as Moletrust and PageRank produce trust score predictions in a continuous interval while others just the 4 discrete trust scores, we decided to apply a rounding to the closest possible certification value before the predicted trust scores are compared with the real values so that for example a predicted trust score of 0.746 becomes 0.8 (Journeyer).

The results of the evaluation are reported in Table 4.1. We start by commenting the column "fraction of wrong predictions". Our baseline is the trust metric named "Random" which produces an incorrect predicted trust score 74% of the times. The best one is Ebay with an error as small as 35% followed by Moletrust2 (36.57%), Moletrust3 (37.60%) and Moletrust4 (37.71%). Increasing the trust propagation horizon in Moletrust allows to increase the coverage but also increases the error. The reason is that users who are nearby in the trust network (distance 2) are better predictors than users further away in the social network (for example, users at distance 4). This is consistent with experiments on other social networks [11].

Note that Moletrust is a local trust metric that only uses information available "near" the source node so it can be run on small devices such as mobiles which only need to fetch information from the (few) trust users and possibly the users trusted by them. This behaviour is tunable through setting the trust propagation horizon to specific values. On the other hand, Ebay, being a global trust metric, must aggregate the entire trust network, which can be costly both in term of bandwidth, memory and computation power. So a local trust metric tends to require less information for producing recommendations which might be a desirable features in some situations.

The AlwaysX metrics depend on the distributions of certifications and are mainly informative of the data distribution.

The fraction of wrong predictions of Advogato (both local and global) is high compared to Ebay and Moletrust. The reason is that Advogato was not designed for predicting an accurate trust value for a specific pair of users (the trust A should place in B) but to increase attack-resistance [8], i. e. being able to exclude malicious users, while accepting as many valid accounts as possible. A side effect is that it limits the amount of granted global certifications and assigns a large number of Observer certificates. In the case of AdvogatoGlobal, 45% of the predicted global certifications are marked as Observer which obviously has an impact on the leave-one-out evaluation. Different trust metrics might have different goals, that require different evaluation techniques. We could have tuned different parameters of Advogato for making it perform differently, however our intention was to evaluate the original trust metric in the task of predicting trust scores so we decide to run Advogato with the original parameters. Note also that the local version of Advogato is more accurate than the global version. The last metric shown in Table 2.1 is PageRank [2]: the fraction of correct predictions is not too high but again the real intention of PageRank is to rank web pages and not to predict the correct value of assigned trust.

An alternative evaluation measure is the Mean Absolute Error (MAE). The MAE is computed by averaging the difference in absolute value between the real and the predicted trust statement on an edge. There is no need to round values to the closest certification value because MAE computes a meaningful value for continuous values. However, in order to fairly compare trust metrics that return real values and trust metrics that return discrete values, we choose to perform anyway the rounding to the closest possible certification value before computing MAE.

The second column of Table 4.1 reports the MAE for the evaluated trust metrics. The baseline is given by the Random trust metric which incurs in a MAE of 0.2230. These results are the worst besides the trivial trust metrics that always predict the most infrequent certification values. Predicting always Journeyer (0.8) incurs in a small MAE because this value is frequent and central in the distribution of assigned trust scores. Ebay is the trust metric with the best performance, with a MAE of 0.0855. And it is again followed by Moletrust that in a similar way is more accurate with smaller trust propagation horizons than with larger ones.

A variant of MAE is Root Mean Squared Error (RMSE). RMSE is the root mean of the average of the squared differences. This evaluation measure tends to emphasize large errors, which favor trust metrics that remain within a small band of error and don't have many outlying predictions that might undermine the confidence of the user in the system. For example, it penalizes a prediction as Observer when the trust score the source user would have assigned was Master, or vice versa. The baseline trust metric Random has an RMSE of 0.2839. Again Ebay is the best metric with an RMSE of 0.1563 and all the other performances exhibit a pattern similar to the one exposed for the other evaluation measures. However there is one unexpected result: the trivial trust metric OutA is the second best, close to Ebay. Remind that, when asked a prediction for the trust user A should place in user B, OutA simply returns the average of the trust statements going out of A, i. e. the average of how user A judged other users. This trust metric is just a trivial one that was used for comparison purposes. The good performance of OutA in this case is related to the distribution of the data in this particular social setting. The Observer certification has special semantics: it is the default value attributed to a user unless the Advogato trust metric gives a user a higher global certification. So there is little point in certifying other users as Observer. In fact, the FAQ specifies that Observer is "the level to which you would certify someone to remove an existing trust certification". Observer certifications are mainly used when a user changes its mind about another user and wants to downgrade her previously expressed certification as much as possible. This is also our reason for mapping it to 0.4, a less than sufficient level. As a consequence of the special semantics of observer certifications, they are infrequently used. In fact only 638 users used the Observer certification at least once while, for instance, 2938 users used the Master certification at least once. Trust metrics like Ebay and Moletrust work doing averages of the trust edges of the network (from a global point of view for Ebay and only considering the ones expressed by trusted users for Moletrust) and, since the number of Observer edges is very small compared with the number of Master, Journeyer and Apprentice edges, these predicted average tend to be close to higher values of trust. This means that when predicting an Observer edge (0.4) they tend to incur in a large error. This large error is emphasized by the squaring of the RMSE formula. On the other hand, using the average of the outgoing trust edges (like OutA does) happens to be a successful technique for not incurring in large errors when predicting Observer edges. The reason is that a user who used Observer edges tended to use it many times so the average of its outgoing edge certifications is a value that is closer to 0.4 and hence it incurs in lower errors on these critical edges and, as a consequence, in smaller RMSE. This effect can also be clearly seen when different trust metrics are restricted to predict only Observer edges and evaluated only on them. In this case (not shown in Tables), OutA gets the correct value for trust (Observer) 42% of times, while for instance, Ebay only 2.7% of times and Moletrust2 4%. The fact the trivial trust metric

Table 4.2
Evaluation of trust metrics on trust edges going into controversial users

	Fraction wrong predictions	MAE	RMSE	Coverage
Random	0.799	0.266	0.325	1.00
AlwaysMaster	0.462	0.186	0.302	1.00
AlwaysJourneyer	0.801	0.202	0.238	1.00
AlwaysApprentice	0.943	0.296	0.320	1.00
AlwaysObserver	0.794	0.414	0.477	1.00
Ebay	0.778	0.197	0.240	0.98
OutA	0.614	0.147	0.199	0.98
OutB	0.724	0.215	0.280	0.92
Moletrust2	0.743	0.195	0.243	0.80
Moletrust3	0.746	0.194	0.241	0.93
Moletrust4	0.746	0.195	0.242	0.95
PageRank	0.564	0.186	0.275	1.00
AdvogatoLocal	0.518	0.215	0.324	1.00
AdvogatoGlobal	0.508	0.216	0.326	1.00

OutA exhibits a so small RMSE supports the intuition that evaluating which conditions a certain trust metric is more suited for than another one is not a trivial task. Generally knowledge about the domain and the patterns of social interaction is useful, if not required, for a proper selection of a trust metric for a specific application and context.

The last column of Table 4.1 reports the coverage of the different trust metrics on the Advogato dataset. For some trust edges, a trust metric might not be able to generate a prediction and the coverage refers to the number of edges that are predictable. The experiment shows that the coverage is always very high. Since local trust metrics use less information (only trust statements of trusted users) their coverage is smaller than the coverage of global trust metrics. Anyway, differently from other social networks [11], it is very high. The Advogato trust network is very dense, so there are many different paths from a user to another user. Even very lo cal trust metrics such as Moletrust2, that only use information from users at distance 2 from the source user, are able to cover and predict almost all the edges.

4.2. Evaluation of trust metrics on controversial users. As a second step in the analysis we devoted our attention to controversial users [11]. Controversial users are users which are judged in very diverse ways by the members of a community. In the context of Advogato, they can be defined as users who received many certifications as Master and many as Apprentice or Observer: the community does not have a single way of perceiving and judging them. The intuition here is that a global average can be very effective when all the users of the community agree that "raph" is a Master, but there can be situations in which something more tailored and user specific is needed, especially when there isn't a subjective judgment that is shared by all the members of the community.

With this in mind we define controversiality level of an Advogato user as the standard deviation in certifications received by that user, similarly to previous studies [11]. The higher the standard deviation, the more controversial the user is. A user with controversiality level as 0 is not controversial at all since all the other users certify her with the same value. The certification level is not very meaningful when the number of received certifications for an user is small (for example 3); for this reason in the following we are going to report measures on users who received at least 10 or 20 incoming certificates, and for which the standard deviation in received certifications really represents the fact the community does not have a single way of perceiving these popular users.

In Table 4.2 we report the evaluation of the performances of the same trust metrics of Table 4.1 but evaluated only on trust edges going to Advogato users with at least 10 incoming edges and controversiality level of 0.2. In this way we reduce the number of edges considered in the evaluation from 52, 981 to 2, 030, which is still a significant number of edges to evaluate trust metrics on. Figure 4.1 graphically reports the number of edges going into users (who received at least 20 certifications) with at least a certain controversiality level for

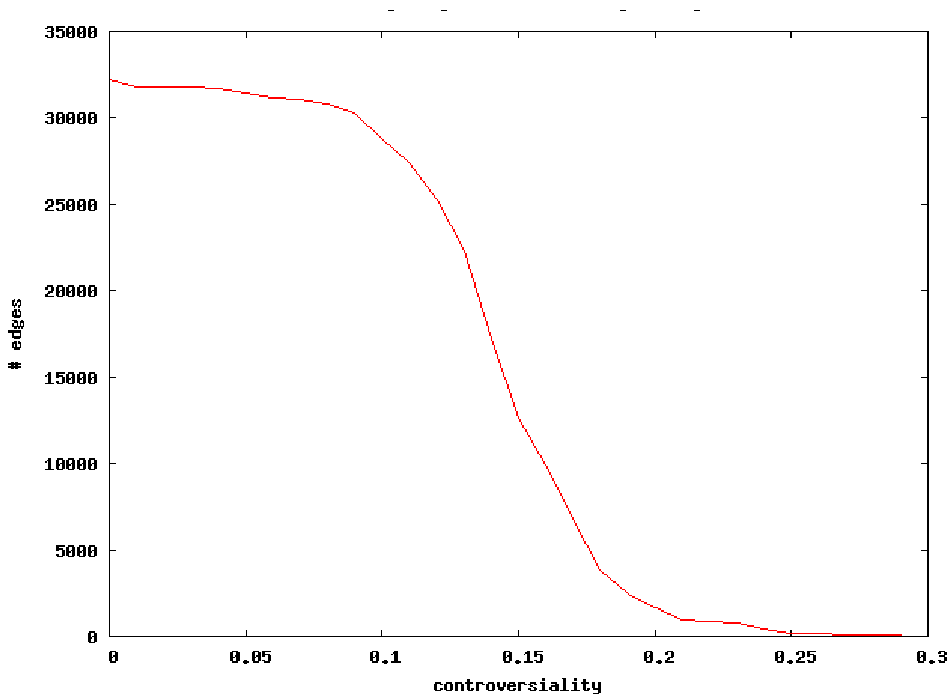


FIG. 4.1. *Number of edges per controversiality level*

all controversiality levels from 0 to 0.3. As intuitive, increasing the controversiality level of users decreases the number of edges going into users with at least that controversiality level.

Figure 4.2 on the other hand shows how at higher controversiality levels the percentage of polarized trust scores increases: certifications as Master and Observer becomes more frequent. This means that predicting trust edges going into controversial users is in theory more difficult, since it is important to predict the correct trust score which is not close to an average score. Both figures confirm intuition and are informative of the distribution of trust scores.

Going back to the evaluation measures presented in Table 4.1, we start by commenting the evaluation measures on AlwaysMaster (second row of Table 4.2) because it presents some peculiarities. AlwaysMaster predicts the correct trust value 53.84% (100% 46.16%) of times and, according to the evaluation measure "fraction of correctly predicted trust statements", seems a good trust metric, actually the best one. However the same trust metric, AlwaysMaster, is one of the less precise when RMSE is considered. A similar pattern can be observed for AdvogatoGlobal. In fact, since in general there is at least one flow of trust with Master certificates going to these controversial users, AdvogatoGlobal tends to predict almost always Master as trust value and since almost half of the edges going into controversial users are of type Master, AdvogatoGlobal often predicts the correct one.

The results presented in Table 4.1 suggest that the same trust metric might seem accurate or inaccurate depending on the choice of the evaluation measure. This fact once more highlights how evaluating trust metrics on real world datasets is a complicated task and a comparison of same trust metrics on many different datasets according to different evaluation methods would be highly beneficial for understanding the situations in which one trust metric is more appropriate and useful than another. We already previously explained why OutA is able to have a so small RMSE, the smallest one on users with controversial level of 0.2: based on how Observer certifications are used in the system, OutA is the only metric that is able to avoid large errors when predicting the Observer edges, which are a relevant percentage since the evaluated users are controversial.

Arriving at a comparison between a global trust metric such as Ebay and a local trust metric such as Moletrust, we were expecting the latter to be significantly more accurate than the first one on controversial users. While on the Epinions dataset, this is what was observed [11], the same is not true here since the two trust metrics incur in very similar performances.

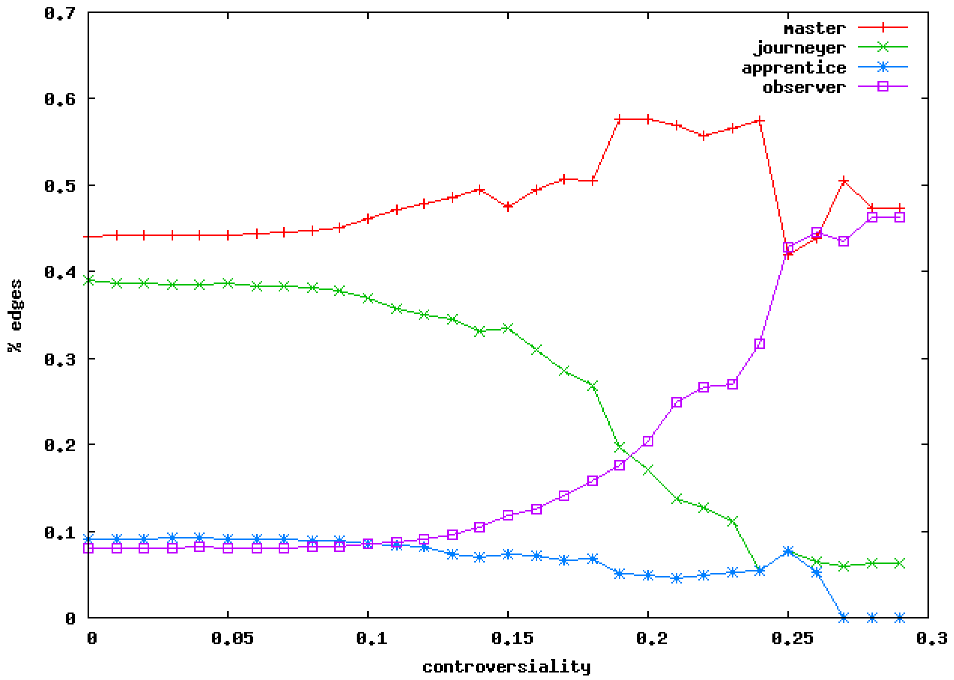


Fig. 4.2. *Percentage of edges for each type per controversiality level*

Figure 4.3 graphically presents the performances (measured by RMSE) of some selected trust metrics on users with increasing controversiality levels and at least 20 incoming edges. It can be observed that the local trust metrics MoletrustX starts to perform better than Ebay and other metrics when the controversiality levels is larger than 0.25. However the difference is not that large as expected.

The reason for this similarity of performances between Ebay and Moletrust2 is partly that in Epinions, the trust values were binary (either trust or distrust) and it was easier to discriminate. Another reason seems to be that on Advogato the user base is not divided in cliques of users such that users of one clique trust each other and distrust users of other cliques. In fact Advogato users are somehow similar and feel part of one single large community. It is future work to analyze if on a social network with a much higher polarization of opinions (such as for example on essembly.com, a political site, in which users tend to express strong feeling for or against other people based on their political views) the performances of lo cal trust metrics are significantly better than global ones. The study on the Advogato trust network dataset presented in this paper does not allow arguing that local trust metrics and in general complex trust metrics are needed in order to outperform simpler trust metrics. Another future work is exploring different evaluation procedures which might be more informative of the real performances of different trust metrics.

5. Conclusions. In this paper we have presented Trustlet [1], an open environment for research on trust metrics. We have claimed that the rapid development of social networking sites [4] asks for a shared effort in collecting datasets and distributing code of algorithms so that comparisons of different research proposals is easier, replicable and more coherent.

As an initial investigation we have reported our comparison of different trust metrics on the Advogato dataset. The results are partly contradictory and this suggests there is need to run systematically evaluations of different algorithms against a large number of different datasets. As future works we are looking into extending our analysis to more datasets also from different social scenarios, for example the networks of relationships (coediting, talk) among Wikipedia users.

Our goal is to make Trustlet an environment which facilitates this collaborative effort. We believe research on these topics is very needed in a time in which our relationships are starting to move more and more into the "virtual" world and our society and life is affected significantly from the predictions and suggestions produced by many different algorithms.

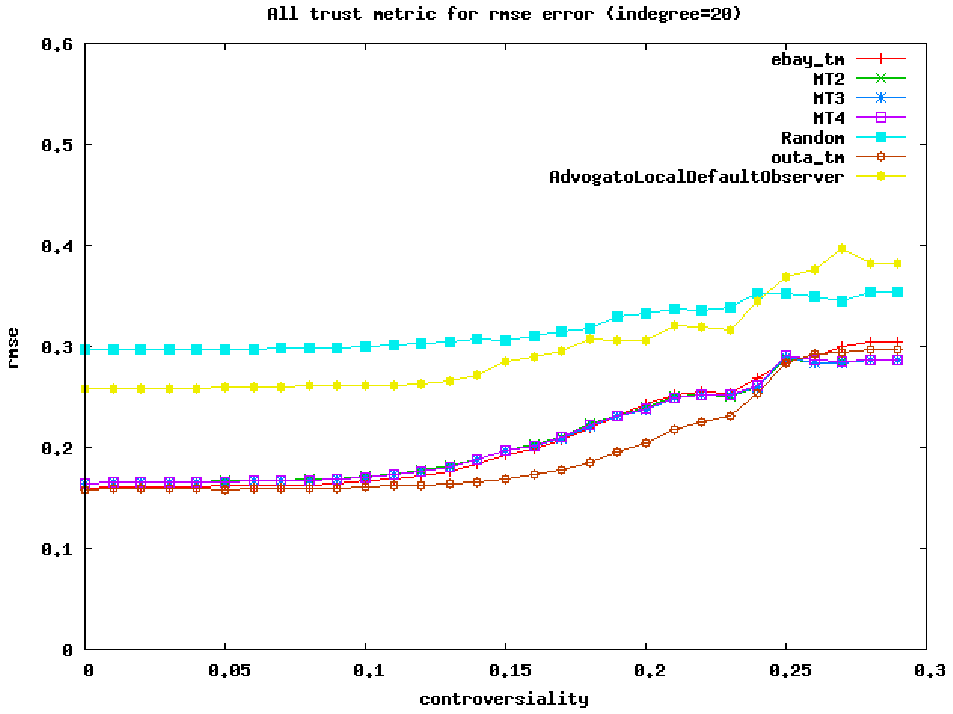


Fig. 4.3. *RMSE for some trust metrics for different controversiality levels*

REFERENCES

[1] *Trustlet, collaborative wiki for trust research.* `http://www.trustlet.org`

[2] D. Austin, *How google finds your needle in the web's haystack.* retrieved on 2008-02-02. `http://www.ams.org/featurecolumn/archive/pagerank.html` 2006.

[3] A.-L. Barabasi, *Linked: The New Science of Networks*, Perseus, Cambridge, MA, 2002.

[4] d. m. boyd and N. B. Ellison, *Social network sites: Definition, history, and scholarship*, Journal of Computer-Mediated Communication, 13 (2007), p. art. 11.

[5] F. Fukuyama, *Trust: the Social Virtues and the Creation of Prosperity*, Free Press Paperbacks, 1995.

[6] K. K. Fullam, T. Klos, G. Muller, J. Sabater-Mir, K. S. Barber, and L. Vercouter, *A specification of the agent reputation and trust (art) testbed*, in Proceedings of 4th AAMAS Conference, Utrecht, 2005.

[7] J. Golbeck, *Computing and Applying Trust in Web-based Social Networks*, PhD thesis, University of Maryland, 2005.

[8] R. Levien, *Attack resistant trust metrics.* Ongoing PhD thesis. `http://www.levien.com/thesis/compact.pdf`

[9] J. Locke, *An Essay concerning Human Understanding*, Harvester Press, Sussex, 1680.

[10] P. Massa, *Trust in E-Services: Technologies, Practices and Challenges*, Idea Group, Inc, 2006, ch. A survey of trust use and modeling in current real systems.

[11] P. Massa and P. Avesani, *Trust metrics on controversial user: Balancing between tyranny of the majority and echo chambers*, International Journal on Semantic Web and Information Systems, 3 (2007), p. 39ij64.

[12] J. Moreno, *Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Sociodrama*, Beacon House, New York, 1953.

[13] M. E. J. Newman, *The structure and function of complex networks*, SIAM Review, (2003), pp. 167–256.

[14] D. Quercia, S. Hailes, and L. Capra, *Lightweight distributed trust propagation*, in Proceedings of the 7th IEEE International Conference on Data Mining, 2007.

[15] J. Sabater and C. Sierra, *Review on computational trust and reputation models*, Artificial Intelligence Review, 24 (2005), pp. 33–60.

[16] C. Ziegler, *Towards Decentralized Recommender Systems*, PhD thesis, Albert-Ludwigs-Universitaet Freiburg, 2005.

Edited by: Dominik Flejter, Tomasz Kaczmarek, Marek Kowalkiewicz
Received: February 9th, 2008
Accepted: March 19th, 2008
Extended version received: August 8th, 2008

ISSN 1895-1767

BOOK REVIEW

EDITED BY SHAHRAM RAHIMI

Introduction to Data Mining
Pang-Ning Tan, Michael Steinbach, and Vipin Kumar
Addison Wesley

This text book is an advanced introduction and reference to data mining from an algorithmic perspective. It is appropriate for graduate students and researchers. An instructor should select parts of the book for a curriculum rather than proceeding straight through from the beginning. The authors suggest even-numbered chapters for classroom instruction and most of these chapters do not have to be covered in order. (The chapter on anomaly detection, however, appropriately needs to be studied last, because it relies on information provided in earlier chapters.)

The explanation of some basic procedures such as C4.5, CART, and Page Rank, are left out. However, other more complicated procedures, such as SVM, are explained in some detail, making this book more appropriate for advanced courses than for introductory ones.

The first three chapters introduce the reader to data mining, data, and, general exploration of data. Data mining is introduced in Chapter 1 in terms of describing data and making predictions from it. Chapter 2 discusses types and quality of data, preprocessing data, and measures of similarity and dissimilarity. Summary statistics, visualization, and On-Line Analytical Processing (OLAP) are major themes covered in Chapter 3 to explore data.

The bulk of the book has two chapters each on classification, association, and clustering, which do not have to be read in order.

Basic concepts of classification are presented in Chapter 4 and include decision tree induction, as well as overfitting, evaluating performance of a classifier, and methods of comparing classifiers. Chapter 5 discusses alternative techniques of classifying, including rule-based classifiers, nearest neighbor classifiers, Bayes, artificial neural networks (ANN), support vector machines (SVM), and ensemble methods. This chapter also covers the issues of class imbalance and multiclasses.

Association analysis begins in Chapter 6 and concentrates on the *Apriori* principal to generate frequent itemsets and rules. Other topics in this chapter are compact representation of frequent itemsets, althernative methods for generating frequent itemsets, the FP-Growth Algorithm, evaluation of association patterns, and the effect of skewed support distribution. Chapter 7 continues with association analysis by considering categorical attributes, continuous attributes, a concept hierarchy, sequential patterns, subgraph patterns, and infrequent patterns.

Cluster analysis begins with basic concepts in Chapter 8 and continues with additional issues and algorithms in Chapter 9. Numerous clustering techniques are discussed, including K-means, agglomerative, DBSCAN, fuzzy, Expectation-Maximization (EM), Self-Organizing Maps (SOM), CLIQUE, DENCLUE, sparsification, Minimum Spanning Tree (MST), OPOSSUM, Chameleon, shared nearest neighbor similarity, Jarvis-Patrick, SSN density, BIRCH, and CURE. These chapters also discuss evaluating clusters and clustering algorithms. The authors note in Chapter 9 that a firm understanding of statistics and probability is required for some of these methods.

The last chapter covers anomaly detection and presumes that the reader is famliar with some of the concepts covered in previous chapters. Statistical, proximity-based, density-based, and clustering techniques are discussed.

Appendices include background information on linear algebra, dimensionality reduction, probability, statistics, regression, and optimization.

The outline of the book is not always parallel in the way that subsections are organized, which can cause confusion to the reader who is attempting to understand the context of what is being presented. For example, Chapter 5 has more than one subsection on types of classifiers, then a subsection each on a specific type of classifier, ensembles of classifiers, and a general problem having to do with classifying. A better outline would have made it more clear when the text was presenting types of classifiers, when it was explaining specfic examples of classifying algorithms, and when it was discussing meta information about classification. An introductory text should lead the student more gently into the maze of data mining concepts and methods.

This book, in summary, is a good reference that provides deeper information about data mining methods than can be easily found elsewhere. It is also appropriate as a supplemental text or for an advanced introductory course.

Chet Langin

AIMS AND SCOPE

The area of scalable computing has matured and reached a point where new issues and trends require a professional forum. SCPE will provide this avenue by publishing original refereed papers that address the present as well as the future of parallel and distributed computing. The journal will focus on algorithm development, implementation and execution on real-world parallel architectures, and application of parallel and distributed computing to the solution of real-life problems. Of particular interest are:

Expressiveness:

- high level languages,
- object oriented techniques,
- compiler technology for parallel computing,
- implementation techniques and their efficiency.

System engineering:

- programming environments,
- debugging tools,
- software libraries.

Performance:

- performance measurement: metrics, evaluation, visualization,
- performance improvement: resource allocation and scheduling, I/O, network throughput.

Applications:

- database,
- control systems,
- embedded systems,
- fault tolerance,
- industrial and business,
- real-time,
- scientific computing,
- visualization.

Future:

- limitations of current approaches,
- engineering trends and their consequences,
- novel parallel architectures.

Taking into account the extremely rapid pace of changes in the field SCPE is committed to fast turnaround of papers and a short publication time of accepted papers.

INSTRUCTIONS FOR CONTRIBUTORS

Proposals of Special Issues should be submitted to the editor-in-chief.

The language of the journal is English. SCPE publishes three categories of papers: overview papers, research papers and short communications. Electronic submissions are preferred. Overview papers and short communications should be submitted to the editor-in-chief. Research papers should be submitted to the editor whose research interests match the subject of the paper most closely. The list of editors' research interests can be found at the journal WWW site (`http://www.scpe.org`). Each paper appropriate to the journal will be refereed by a minimum of two referees.

There is no a priori limit on the length of overview papers. Research papers should be limited to approximately 20 pages, while short communications should not exceed 5 pages. A 50–100 word abstract should be included.

Upon acceptance the authors will be asked to transfer copyright of the article to the publisher. The authors will be required to prepare the text in $\LaTeX\,2_\varepsilon$ using the journal document class file (based on the SIAM's `siamltex.clo` document class, available at the journal WWW site). Figures must be prepared in encapsulated PostScript and appropriately incorporated into the text. The bibliography should be formatted using the SIAM convention. Detailed instructions for the Authors are available on the SCPE WWW site at `http://www.scpe.org`.

Contributions are accepted for review on the understanding that the same work has not been published and that it is not being considered for publication elsewhere. Technical reports can be submitted. Substantially revised versions of papers published in not easily accessible conference proceedings can also be submitted. The editor-in-chief should be notified at the time of submission and the author is responsible for obtaining the necessary copyright releases for all copyrighted material.