

From OpenSocial to Enterprise 2.0 based on Social Software Concepts and Technologies

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

Oliver Grof

Matrikelnummer 0227374

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuerin: Assoc. Prof. Dipl. Ingⁱⁿ Drⁱⁿ Hilda Tellioglu

Wien, 26.07.2010

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Erklärung zur Verfassung der Arbeit

Oliver Grof

H-8900 Zalaegerszeg Jokai Mor Str. 23

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Kurzfassung

In den letzten Jahren wurde viel über soziale Software im Unternehmen diskutiert und seit 2006 wird der Begriff Enterprise 2.0 zunehmend benutzt. Die vorliegende Diplomarbeit untersucht die Herkunft dieser Konzepte, stellt eine Verbindung zu anderen wissenschaftlichen Disziplinen her und analysiert die Zukunftsmöglichkeiten. Die Arbeit besteht aus einem theoretischen und einem praktischen Teil. Im theoretischen Teil werden sowohl die aktuellen Trends bezüglich Enterprise 2.0, als auch die zugrunde liegenden Technologien vorgestellt. Weiters werden einige Fallbeispiele und mögliche Probleme hinsichtlich der Benutzbarkeit erörtert. Da speziell soziale Netzwerke an Bedeutung gewinnen, wird im praktischen Teil die Programmierschnittstelle OpenSocial untersucht. Um herauszufinden, inwiefern OpenSocial für die Lösung unternehmensspezifischer Probleme geeignet ist, wurde eine einfache Applikation, der Social Conference Manager, implementiert. Die Applikation wurde in einem offenen sozialen Netzwerk installiert, sodass sie von einigen ausgewählten BenutzerInnen im Rahmen eines Interviews getestet werden konnte. Das Interview beinhaltete sowohl allgemeine Fragen zu Enterprise 2.0, als auch konkrete Fragen über das getestete Tool. Basierend auf den gewonnenen Erkenntnissen werden in der vorliegenden Arbeit einige neue Anforderungen formuliert. Schließlich wird die Frage, ob OpenSocial für den Einsatz im Unternehmen geeignet ist, beantwortet, und es werden auch die Zukunftsperspektiven gezeigt. Insgesamt sind die aktuellen Ansätze aus technologischer Sicht vielversprechend; ohne eine reformierte Geschäftskultur und eine gesteigerte Sicherheit jedoch nicht erfolgreich einsetzbar.

Schlüsselwörter: Enterprise 2.0, soziale Software, OpenSocial, soziales Netzwerk

Abstract

In recent years, social software in the enterprise has been increasingly discussed and as of 2006 the term Enterprise 2.0 has been increasingly used. The present thesis explores the origins of these concepts, connects to other scientific disciplines and analyses future possibilities. The work consists of a theoretical and a practical part. The theoretical part introduces both the current trends with respect to Enterprise 2.0 and the underlying technologies. Furthermore some case studies and usability issues are discussed. Since social networks gain in importance, the OpenSocial API is studied in the practical part. To determine whether OpenSocial is a suitable solution for enterprise-specific problems, a simple application, namely Social Conference Manager, has been implemented. The application has been added to an open social network, so that selected users were able to test it within an interview. The interview contained both general question related to Enterprise 2.0 and concrete questions about the tool, that had been tested. Based on the lessons learned, some new requirements are formulated in the present work. Finally, the question of whether OpenSocial is suitable for use in the enterprise is answered, and also the future prospects are shown. Overall, the current approaches are promising from a technological point of view; however, they cannot be used successfully without a reformed business culture and without an enhanced security.

Keywords: Enterprise 2.0, social software, OpenSocial, social network

Contents

1	Introduction	1
2	Related Research	3
2.1	Groupware and CSCW	4
2.2	Business Process Management	6
2.3	Knowledge Management	8
2.4	Online Communities: Web 2.0	10
2.5	Enterprise Mashups	14
3	Enterprise 2.0 Concepts and Trends	17
3.1	SLATES and FLATNESSES	18
3.2	Software Solutions	19
3.2.1	Sharepoint and Enterprise 2.0	20
3.2.2	Case Studies	22
3.3	Usability Issues	28
3.4	Underlying Technologies	30
3.5	The Influence of CSCW and Web 2.0	35
3.6	Security and Privacy	37
4	OpenSocial™ in Action	38
4.1	Application Development	39
4.2	Deployment in Sandboxes	44
4.3	Social Conference Manager	46
4.3.1	Use Cases	47
4.3.2	Implementation Details	53
5	Interview Analysis	60
5.1	Derived Requirements	61
5.2	Opportunities for Further Development	64

6 Latitude for Enterprise 2.0	67
6.1 OpenSocial in the Enterprise	67
6.2 Alternative Solutions	72
6.3 Social Semantic Web	74
6.4 Assuming Responsibility	79
7 Summary	84
A User Interviews	88
B JSON Structures	97
Bibliography	99
Index	105

List of Figures

2.1	Classification of groupware based on the time/space matrix . . .	4
2.2	Classification of groupware based on interaction types	5
2.3	Prominent BPM standards	7
2.4	Social-Software-Triangle	13
2.5	Resources consumed by enterprise mashups	15
3.1	FLATNESSES	19
3.2	Capability areas of SharePoint 2010	23
3.3	Examples of site content, categorized by users goals of caring, climbing, or campaigning	25
3.4	REST Example 1: creating a user with HTTP POST	32
3.5	REST Example 2: update a user with HTTP PUT	32
3.6	Example of a valid JSON	33
3.7	Example of a valid JSON completed with types	33
3.8	Differences between groupware and social software	36
4.1	Application architecture	39
4.2	Use case diagram of the Social Conference Manager	47
4.3	Application architecture (orkut)	54
4.4	Database structure	55
4.5	The Social Conference Manager developed for this thesis	56
4.6	The table deleteCount	58
6.1	An RDF-triple as a directed graph	76

Listings

4.1	A basic Google Gadget created for this thesis	41
4.2	A basic OpenSocial application created for this thesis	43
4.3	The profile view of the Social Conference Manager developed for this thesis	59
6.1	XML representation of the conference data	77
6.2	RDF representation of the conference data	78
6.3	Exemplary RDF schema file developed for this thesis	79

Abbreviations

AJAX Asynchronous Javascript and XML

API Application Programming Interface

BPM Business Process Management

CSCW Computer Supported Cooperative Work

CSS Cascading Style Sheets

DOM Document Object Model

E2.0 Enterprise 2.0

e.g. *exempli gratia*

etc. *et cetera*

HTTP HyperText Transfer Protocol

i.e. *id est*

IP Internet Protocol

IT Information Technology

JDK Java Development Kit

JSON JavaScript Object Notation

KM Knowledge Management

URI Uniform Resource Identifier

UWA Universal Widget API

PHP Hypertext Preprocessor

REST Representational State Transfer

RDF Resource Description Framework

RSS Really Simple Syndication

SDN SAP Developer Network

XHR XMLHttpRequest

XHTML Extensible Hypertext Markup Language

XML eXtensible Markup Language

Chapter 1

Introduction

Communication, interaction and the close collaboration of employees are crucial for the success of an enterprise. Information technology, that supports cooperative work adequately, gains in importance with the increasing size of a company and the increasing complexity of its organisational structure. One approach for IT-supported cooperative work, which has been increasingly discussed in recent years, is the adoption of *social software* in an enterprise. As of 2006 this concept is, inspired by the success of various Web 2.0 platforms, called *Enterprise 2.0* [35]. However, a consistent, standardized definition does not exist. Therefore I will give an overview of both the theoretical and the technical background of Enterprise 2.0. The theoretical part focuses on results and findings of publications, conferences and blogs. The technical part addresses the OpenSocial API, which was released by Google on November 1, 2007 in order to support the development of web-based social network applications. Overall, I examine the applicability of this particular technology for Enterprise 2.0.

The thesis strongly relies on the exhaustive literature research, that has been done within the scope of the present work. In addition, I have implemented a prototype application, which is evaluated based on user interviews.

The structure of this work is as follows.

Chapter 2 introduces the current state of related research and clarifies important terms related to the present thesis. After a general overview, CSCW¹, online communities, knowledge management and business process management are discussed in detail. The explanation of these fun-

¹Computer Supported Cooperative Work. See 2.1 for more details.

damental terms is supposed to show the origins of Enterprise 2.0 as well.

Chapter 3 discusses concepts, trends and issues concerning the adoption of social software in enterprises. The chapter's focus is on the findings of divers publications. Besides the general discussion of Microsoft Sharepoint, concrete Enterprise 2.0 solutions are examined based on selected case studies. Furthermore usability issues and underlying technologies are presented.

Chapter 4 introduces Google's OpenSocial API and presents the social software implemented within the scope of this master's thesis. Since OpenSocial is a relatively new technology for network programming, the fundamental characteristics are described based on two basic examples first. Second, different sandboxes for development are evaluated. Finally, the application called Social Conference Manager is documented.

Chapter 5 evaluates the interviews done within the scope of the present thesis, deduces new requirements and presents opportunities for further development. The chapter explains the guidelines and the course of the user interviews, and contains the analysis and the prototype evaluation.

Chapter 6 describes the potentials of OpenSocial with respect to Enterprise 2.0. The chapter covers four relevant topics. Besides the general analysis of the relationship and the possibilities, alternative solutions and the social semantic web are discussed. Finally, some risks are pointed out, because the available technologies have to be used the responsible way.

Chapter 7 gives a short summary and suggestions for further research.

Chapter 2

Related Research

In recent years the application of social software in business context has been increasingly discussed and several studies have been carried out [38]. However, the origins of Enterprise 2.0 are concepts described by terms like EIES¹, groupware and CSCW². In fact the scientists Vannevar Bush, Douglas Engelbart and Joseph Carl Robert Licklider predicted already in the 1940s and 1960s that computers will not only be useful for data processing but also for communication and coordination [71]. In the 1980s and 1990s groupware and CSCW became popular terms; the term “groupware”, for example, was shortly adopted by the EIES community [9]. One challenge of cooperative work is “to find out how computer systems can support coordination activities” [54, p.16]. The section “Coordination Support of CSCW Systems” in [54, p.30] gives a very good summary of research done in this area. Nevertheless, both a computer supported knowledge management and an adequate business process management are essential requirements for the efficient coordination of business projects, and need to be integrated in the cooperative work environment [56] [27].

The significant global upturn of the Internet in the mid-nineties opened the doors for a revised view of distributed systems, and brought new opportunities for companies to coordinate the employees’ collaboration. Most of the terms related to social software originate from the early 21st century; wikis and blogs, for instance, have been associated with social software since 2002 [67]. In 2006 Andrew P. McAfee introduced “Enterprise 2.0” [35] motivated by Web 2.0 concepts. Although Web 2.0 is considered as a marketing buzzword, the term itself can be found in a number of scientific papers, e.g., in the prob-

¹Electronic Information Exchange System. The first major implementation of collaborative software [9].

²Computer supported cooperative work

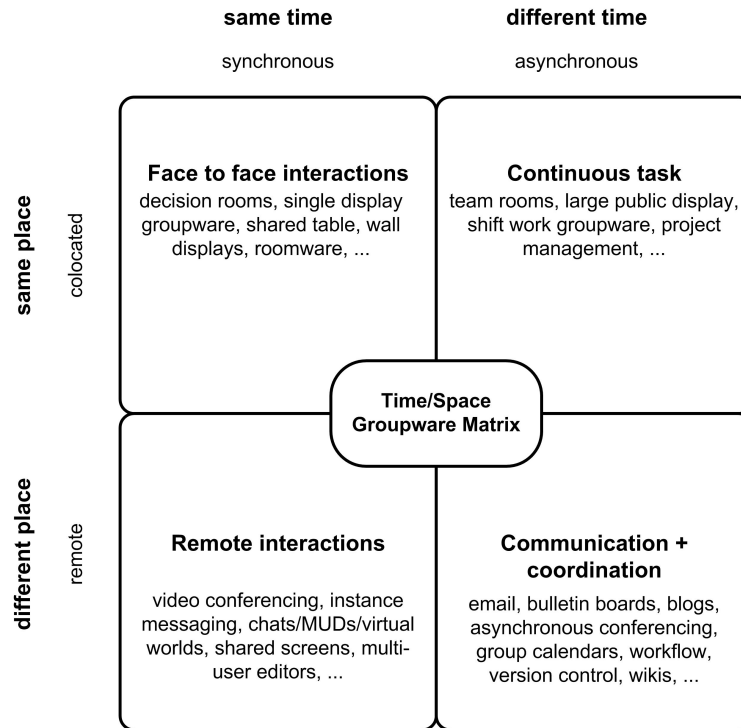


Figure 2.1: Classification of groupware based on the time/space matrix [65]

ably best-known publication in this field written by Tim O’Reilly [48] The article “Enterprise 2.0: The Dawn of Emergent Collaboration” [35] describes technologies, ground rules, challenges and opportunities of social software in industrial context, and is therefore one of the most cited resource in this research area. Further papers examine “the potential of social software from the standpoint of industrial employment” [29, p.381], the motivation of users [26], usability issues or analyse open social networks [42].

The present chapter contains the definitions of the related terms, and discusses the most important characteristics of the concepts behind them. Chapter 3, on the other hand, addresses Enterprise 2.0 concepts and trends.

2.1 Groupware and CSCW

C.A. Ellis defined groupware as “computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment” [13, p.40]. Accordingly groupware is specified as technologies and tools for supporting communication, cooperation and coordi-

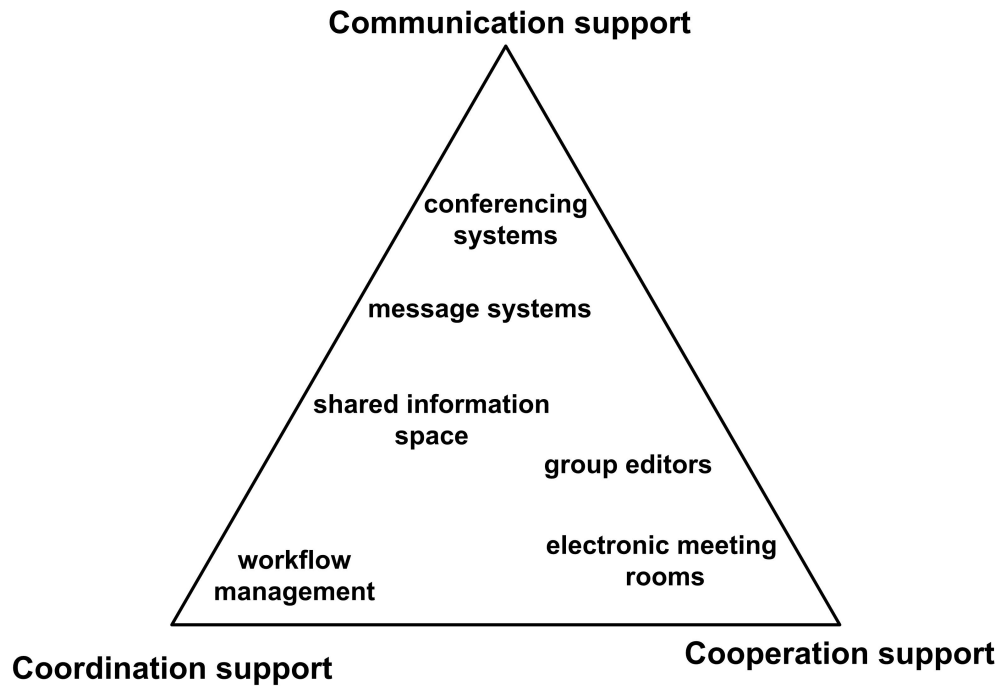


Figure 2.2: Classification of groupware based on interaction types [38, p.18]

nation of activities in groups. The goals are to reduce users' isolation and to provide a high adaptability, since users often find more adequate solutions, if they are able to customize their environment themselves [71, p.45-46].

Computer Supported Cooperative Work (CSCW) is an interdisciplinary field of research studying "tools and techniques of groupware as well as their psychological, social, and organizational effects" [65]. There are two main issues related to CSCW and groupware that must be clarified to get a consistent definition. First, there is a discussion about the relation of these terms. The most cited publications show that the findings of CSCW are the theoretical fundament of groupware, therefore these terms cannot be considered as synonyms. Second, the term "groupware" is used inaccurately due to marketing strategies. Authors of specialised literature (e.g., [71] and [9]) claim that IBM and Microsoft corrupted the term by calling Lotus Notes and Microsoft Exchange/Outlook so.

Groupware can be classified in different ways. Two accepted classifications are the groupware matrix (Figure 2.1) and the groupware triangle (Figure 2.2). According to the matrix, groupware can be divided into two groups, depending on whether the participants are remotely available or colocated. Furthermore

communication can be synchronous or asynchronous. The present thesis focuses on web-based (i.e. remote) groupware, but does not clearly distinguish between synchronous or asynchronous communication. The classification in the triangle, however, is based on the interaction types communication, coordination and cooperation. Especially workflow management is an intensively discussed research area. Workflow management can be viewed as a technology, which supports the process-oriented management discipline business process management (BPM) [27]. Furthermore, Enterprise Resource Planning (ERP) systems are also workflow-supported enterprise software systems. It is important to note that “ERP systems are large and complex systems, which deeply modify the activities and organisation of the companies in which they are implemented” [18, p.10]. Due to the fact that such coordination approaches are essential for successful enterprises, Section 2.2 explains the most important facts related to BPM and the relationship between BPM and ERP.

When discussing computer supported cooperative work, *artifact* is an essential concept. “Artifacts are the main resources hosting information, triggering action, documenting processes, and maintaining knowhow and experiences” [55, p.1]. Section 2.3 addresses the role of artifacts in knowledge management as well.

2.2 Business Process Management

Whenever a large group of people is supposed to collaborate within an enterprise or across enterprises, there is an essential need for appropriate coordination patterns. In this respect, the terms business process and business process management are widely used in industry, therefore they are clarified below.

Business process Business processes are a collection of activities performed by a number of “people/machines/systems from different organizations, working together to achieve a common business goal” [27, p.12]. Referring to [8] a workflow (the basic concept for Workflow Management Systems) is an abstraction of a business process. The “libraries of business processes included in the ERP packages are supposed to make possible the adoption of “best practices” allowing improvements of company performance” [18, p.1].

Business process management BPM supports business processes “using methods, techniques and software to design, enact, control and analyze

	BPM/ SOA/ B2B	Background	Theory/Graphical/ Interchange/Execution Diagnosis/ B2B Info Exchange	Standardized?	Current Status
BPDM	BPM	Industry	Interchange	Yes	Unfinished
BPEL	BPM	Industry	Execution	Yes	Popular
BPML	BPM	Industry	Execution	Yes	Obsolete
BPQL	BPM	Industry	Diagnosis	Yes	Unfinished
BPRI	BPM	Industry	Diagnosis	Yes	Unfinished
ebXML BPSS	B2B	Industry	B2B Info Exchange	Yes	Popular
EDI	B2B	Industry	B2B Info Exchange	Yes	Stable
EPC	BPM	Academic	Graphical	No	Legacy
Petri Net	All	Academic	Theory/Graphical	N.A.	Popular
Pi-Calculus	All	Academic	Theory/Execution	N.A.	Popular
Rosetta-Net	B2B	Industry	B2B Info Exchange	Yes	Popular
UBL	B2B	Industry	B2B Info Exchange	Yes	Stable
UML AD	BPM	Industry	Graphical	Yes	Popular
WSCI	SOA	Industry	Execution	Yes	Obsolete
WSCL	SOA	Industry	Execution	Yes	Obsolete
WS-CDL	SOA	Industry	Execution	Yes	Popular
WSFL	BPM	Industry	Execution	No	Obsolete
XLANG	BPM	Industry	Execution	No	Obsolete
XPDL	BPM	Industry	Execution/Interchange	Yes	Stable
YAWL	BPM	Academic	Graphical/Execution	No	Stable

Figure 2.3: Prominent BPM standards
[27]

operational processes involving humans, organizations, applications, documents and other sources of information” [59, p.4].

In some contexts BPM is misleadingly used interchangeably with SOA³. On the one hand both concepts deal with certain types of coordination, on the other hand “BPM is a process-oriented management discipline aided by IT, and SOA is an IT architectural paradigm” [27, p.16]. Nevertheless, SOA is relevant for the present thesis as well, and additional details are discussed in Section 2.5. As it shown in Figure 2.3, a number of standards have been developed for both BPM and SOA. Currently the Business Process Execution Language (BPEL), the activity diagrams of the Unified Modeling Language (UML AD) and the Web Services Choreography Description Language (WS-CDL) are popular in industry. BPEL and WS-CDL are XML-based languages

³Service-oriented Architecture, set of design principles for decomposition and integration of software services [51, p.276]. Services in SOA can be considered as business tasks, which “are implemented in an environment that facilitates loose coupling with other services” [47, p.22].

for describing business processes and peer-to-peer communication; in contrary, the activity diagrams of the Unified Modeling Language is a graphical modeling language. In the academic context the Petri Net [49] is often cited. The Petri Net is a mathematical modeling language that offers a graphical notation as well.

Marjanovic et al. describe the relation of BPM, CSCW and Knowledge Management:

“In very recent times, organisations have started to shift their focus from highly standardised operational business processes (BPs) to other types of processes that cannot be easily replicated due to the knowledge, skills and creativity of people involved. (...) The renewed interest in process-related knowledge and collaboration has opened a new case for possible synergy of BPM and CSCW (...) [T]he key to this synergy is in the field of Knowledge Management.” [34, p.448]

2.3 Knowledge Management

Knowledge Management (KM) “is the systematic, explicit, and deliberate building, renewal, and application of knowledge to maximize an enterprise’s knowledge-related effectiveness and returns from its knowledge assets” [32, p.1-6] [64]. Technical literature discusses many different aspects of various disciplines; however, two basic orientations of KM are the human-centric and the techno-centric approaches. Publications of recent years introduce integrated approaches, which affiliate the creative, intellectual abilities of individuals and the information processing capacities of computer technologies [16, p.33-34]. Independently of diverse models the exact appreciation of relevant knowledge is crucial for successful knowledge management in enterprises [16, p.37]. On the one hand, the majority of organisation members is familiar with standards, guidelines, business rules, workflow descriptions, information about trends and new technologies etc., more precisely an organization has collective knowledge [16, p.52]. On the other hand, a high amount of relevant information exist in enterprises, which are difficult to capture: for instance, the employee’s knowhow, experiences and their relations within and across companies. Nardi et al. have found that “people invest considerable effort in maintaining links with networks of colleagues, acquaintances, and friends, and

(...) these networks are a significant organizing principle for work and information” [43, p.89]. In this regard the term *personal network* is used. Personal network “is a new paradigm used to describe pervasive computing with strong focus on the person” [5, p.12]. A social desktop called ContactMap has been developed to extract and visualize the information in personal networks [43].

The goal of KM is to collect, store and harness an organisation’s knowledge. Due to the definition of Duncan and Weiss the organisational knowledge base is the collectivity of cooperative mediated knowledge and expertise within an organization [12, p.86f]. Organisations can increase efficiency by enhancing their knowledge base. However, the definition of organisational memory below shows that a positive contribution is not a necessary outcome. “Organisational Memory is the means by which knowledge from the past is brought to bear on present activities, thus resulting in higher or lower levels of organisational effectiveness” [52, p.22].

Computer Supported Solutions

Due to the complexity of knowledge management tasks, it is unavoidable for competitive corporations to use computer supported solutions. A wide range of software products have been released in previous years, which can be grouped as follows [16, p.217ff]:

Content-centric systems Information in documents, pictures etc. need to be managed through the complete lifecycle. Document management systems, content management systems and portals are all content-centric systems.

Artificial intelligence systems Although no unambiguous definition exists, the goal of artificial intelligence approaches is the simulation of human intelligence [16, p.163]. Expert systems, agent systems, text mining systems are artificial intelligence systems.

Management information systems support the management in unstructured (or poorly structured) tasks [16, p.238].

Search services retrieve relevant information.

Groupware and social software are discussed in other sections in detail (Section 2.1, Section 2.4, Chapter 3)

Although these tools are very useful for specific tasks, the optimal solution is the adoption of “complete” KM systems. A knowledge management system is a software system that provides functions for the identification, acquisition, development, distribution, assessment of knowledge and supports organisational learning and organisational effectiveness [16, p.248-249]. In such a system a number of actors operate on common artifacts, which “host valuable information about processes, people’s knowledge, experiences, and habits, conventions and decisions made” [55, p.5]. Besides the concrete content these artifacts contain useful meta-level information like access time, the type of modification, project team constellation), which also have to be made permanent [55]. Hilda Tellioglu has introduced a new approach for knowledge management, that is based on snapshots [56]. Capturing all the relevant information by creating snapshots of the system helps to understand how the artifacts have evolved [56].

2.4 Online Communities: Web 2.0

Various disciplines (e.g., sociology, psychology, anthropology) have already analysed the characteristics of communities, and these analyses are an essential topic of Enterprise 2.0 research as well. Due to the global upturn of the Internet and the advancement of web technologies a number of online communities have arisen. In the first decade of the new millennium the marketing buzzword Web 2.0 has become very popular for embracing such new, social approaches. The present section clarifies some core definitions related to Web 2.0.

Web 2.0 is a term introduced by the web pioneers Tim O’Reilly and Dale Dougherty. They described Web 2.0 as a concept, that has a gravitational core, but not hard boundary [48]. The following traits can be identified based on [48]:

- defining web as a platform
- high increase of user-generated-content, harnessing collective intelligence
- focus on data is crucial for success
- rich user experience

Although the term Web 2.0 is considered as a marketing buzzword by some, it is widely used in research papers as well. However, Sir Timothy John Berners-

Lee⁴ said that Web 2.0 “means using the standards which have been produced by all these people working on Web 1.0” [4], therefore the versioning does not make sense. A number of specialists try to avoid the term by using another, more precise designation (e.g., *Social Web*, *Remixable Web*) for the concepts and technologies they are investigating [17, p.38]. Especially the term Social Web seems to be appropriate for the present work, due to following reasons:

1. The thesis has a special focus on social behaviour and social interactions of employees, which is a foundation of a successful enterprise.
2. The thesis addresses the topic “Social Semantic Web”, which is more easily comprehensible if the term Social Web is used.

To be consistent within the present work and with the literature used in it, the term Web 2.0 is only replaced with Social Web in Section 6.3. It is a fact that the Web has progressed significantly in the last two decades: the so-called Web 2.0 really differs from the so-called Web 1.0 in some aspects. Initially the Web was used to make some pieces of information available and the web content was published and linked by a relatively small number of contributors. In contrast, Web 2.0 is characterised by many participating users, a high amount of user-generated-content and rich user experience. Although several approaches for creating rich user experience have been introduced in the nineties (applets in the Viola browser, Java applets, JavaScript and then DHTML) [48], “the potential of the web to deliver fully scale applications didn’t hit the mainstream till Google introduced (...) web based applications with rich user interfaces and PC-equivalent interactivity” [48]. The term that has grown together with Web 2.0 is AJAX, which is described in detail in Section 3.4. AJAX comprehends a number of technologies, “each flourishing in its own right, coming together in powerful new ways” [48]. The Web has become more dynamic and more personalised: every user can publish own content, use desktop-like applications and even combining content for personal needs is not impossible anymore.

Mashup is a widespread buzzword in the context of state-of-the-art web projects. The term stands for Web 2.0 technologies that combine content from different sources dynamically [17, p.741]. The goal is to create meaningful mashups that create additional value, therefore any other combination of data etc. from different resources should not be forced.

⁴“(...) a British engineer and computer scientist and MIT professor credited with inventing the World Wide Web, making the first proposal for it in March 1989” [Wikipedia]

There are four possible ways of merging content in a web application [17, p.741ff]:

- RSS and Atom access: for more information see 3.4
- Widgets usage: combining both external content and interactive elements (mostly AJAX based) with *Window Gadget*.
- Web scraping: reading, extracting data from web documents
- Accessing data over application programming interfaces (APIs)

Users can merge applications themselves directly in the web or create custom applications tailored to their needs. David Berlind, executive editor of ZDNet, described the analogy of web applications - especially mashups - and desktop applications⁵. Desktop applications run on a certain operating system, like Windows and use APIs for the display or for filesystem access etc. Modern web applications, on the other hand, are Internet-based, run in a browser and use APIs from Google, Yahoo, Amazon etc. According to David Berlind, the big advantage of the Internet is that any developer can publish APIs and a large number of users can create mashups. He calls mashups “the fastest growing ecosystem”.

Certainly, the Internet has evolved technically since 1991, but the sociological changes are particularly striking. A range of *social software* has been created, so the human component has gained in importance and online communities have arisen.

Social software is mostly defined as a subclass of Web 2.0 [71], that “can be loosely defined as software which supports, extends, or derives added value from, human social behaviour” [10]. The social-software-triangle in Figure 2.4 specifies three main concepts how social software can be applied.

Typical examples of social software are weblogs (or blogs) and wikis, but even older concepts (e.g., bulletin boards and instant messaging) can be considered as such. The first blogs have been created in the late nineties [57], and innovations like permalink, trackback and feeds turned blogs “from an ease-of-publishing phenomenon into a conversational mess of overlapping communities” [57, p.20]. Ideal blogs and wikis grant every user easy access, so that

⁵ZDNet video feature: “What is a mashup?” http://news.zdnet.com/2422-13569_22-152729.html (Last visited on 19th January 2010)

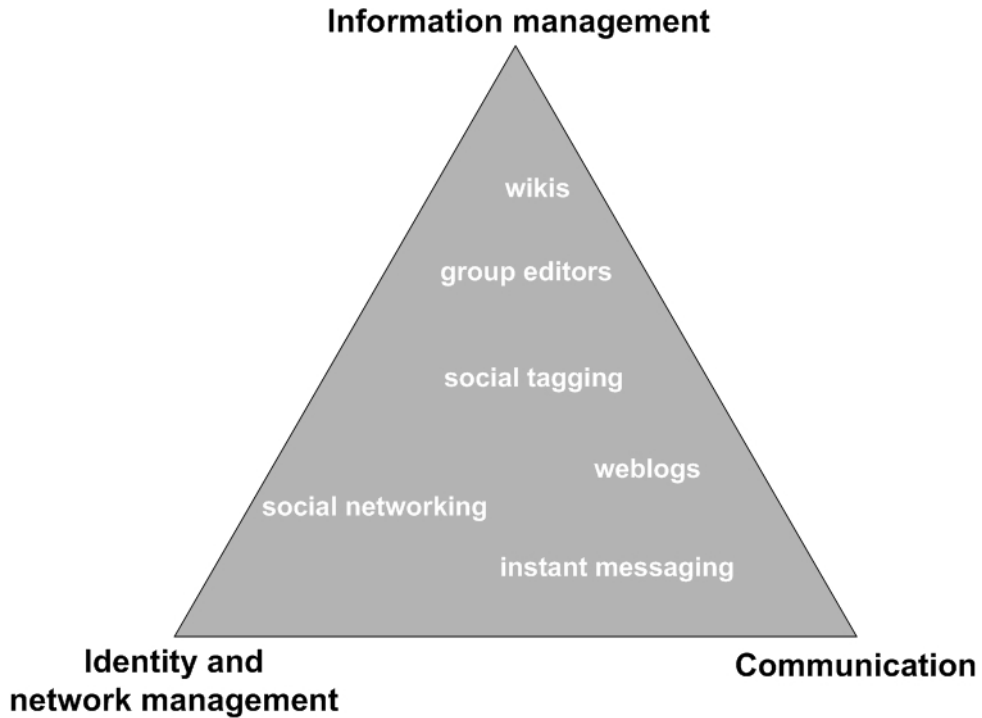


Figure 2.4: Social-Software-Triangle
[71, p.51]

they are not hindered from participating and creating new content. In general, “social software creates its own feedback loop; by building tools that allow people to come together and find each other’s ideas (...), which in turn will bring even more collaboration, cooperation, and conversation online” [57, p.23]. Although social software should have as few limitations as possible, even non-structured organisations are not free of rules: there might exist best practices like posting rules and formatting rules [11]. In some scenarios it might advantageous to share best community practices and to personalise content; however, this should happen with minimal intrusiveness and without defining hard constraints [11]. A possible solution are Rule-based, User-defined Annotations (RU-Annotations) introduced in [11].

As the social software triangle in Figure 2.4 shows, the majority of the tools focus on information management and communication, except *social networking*, which focuses on identity and network management.

Social network service is a form of social software that focuses on building online communities [66]. Participating users publish profiles and create links to any other users with whom they associate, therefore they create

a social network [42].

A number of open social networks exist these days: MySpace⁶, LinkedIn⁷, hi5⁸ and orkut⁹, for instance. However, the most popular social network is probably Facebook¹⁰. Millions of people have created a profile in one or more networks, and there is an increased use of applications available for certain platforms. Some social networks (e.g., Facebook) have released portal-specific APIs, but the applications, that were implemented with this API, had the disadvantage of not being portable. Therefore Google developed OpenSocial to have a standard API for multiple social networks. The main focus of this work is on social network sites that support application development with the OpenSocial API.

2.5 Enterprise Mashups

Due to the expanding use of mashups in business environment Volker Hoyer and Marco Fischer have published a market overview of Enterprise Mashup tools. The literature review within the scope of their paper has resulted in following definition:

“An Enterprise Mashups is a Web-based resource that combines existing resources, be it content, data or application functionality, from more than one resource in enterprise environments by empowering the actual end-users to create and adapt individual information centric and situational applications.” [25, p.710]

Figure 2.5 shows some exemplary resources, that can be consumed and combined. Data quality, reliability, performance and scalability are extremely important requirements for enterprise mashups, as the company processes could be affected¹¹. Interestingly, information technology in the company plays an altered role in this context. Although IT is responsible for the provision and reliable operation of the platform, on that the mashups are created and executed, it seldom affects the implemented functions and operations. The specification of appropriate interfaces, which allow the efficient combination of all the

⁶<http://www.myspace.com/>

⁷<http://www.linkedin.com/>

⁸<http://hi5.com/>

⁹<http://orkut.com/>

¹⁰<http://www.facebook.com/>

¹¹http://wiki.computerwoche.de/doku.php/web_2.0/enterprise-mashups

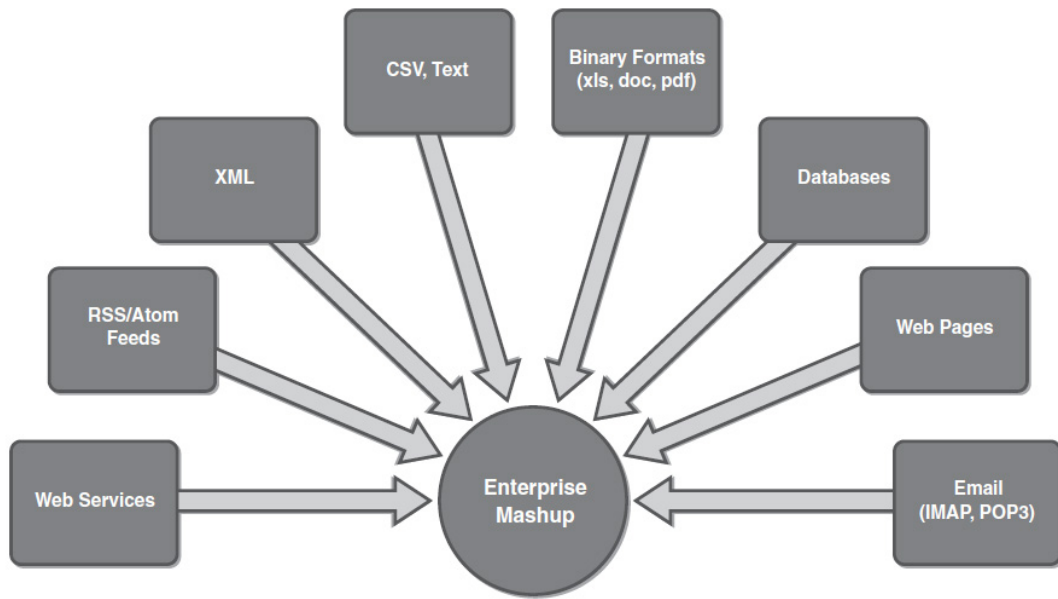


Figure 2.5: Resources consumed by enterprise mashups
[47, p.17]

potential resources, is a fundamental requirement, when designing enterprise mashups. A number of websites (e.g., netvibes¹² and iGoogle¹³) allow users to personalise their start page by using RSS and Atom (see Section 3.4 for more details), or combine content from other sources. Applications, on the other hand, often consume web services described by the Web Service Description Language¹⁴.

Service-Oriented Architecture and Software as a Service

The objective of the present section is to clarify, how the concepts Service-Oriented Architecture (SOA) and Software as a Service (SaaS) are connected to Enterprise Mashups. The term Service-Oriented Architecture stands for a set of design principles for decomposition and integration of software services [51, p.276]. Services in SOA can be considered as business tasks, which “are implemented in an environment that facilitates loose coupling with other services” [47, p.22]. Enterprise mashups (and mashups in general) can combine a range of loose coupling services, because they are based on open standards (e.g., XML, WSDL, UDDI, SOAP) [47, p.22f]. Hoyer and Fischer interpreted En-

¹²<http://www.netvibes.com/>

¹³<http://www.google.com/ig>

¹⁴<http://www.w3.org/TR/wsd1>

terprise Mashups as an evolution of Service-Oriented Architecture(SOA) [25]. However, the focus of SOA patterns is on more abstract, architectural discussions, while Mashup patterns focus on more practical issues [47, p.23-24]. Software as a Service (SaaS) is a business model, whereas “businesses do not invest money to develop and host applications internally, but instead rent the functionality they need from an external service provider” [47, p.25]. On the one hand reliable external providers might reduce security risk, on the other hand some service level agreements have to be negotiated due to possible the lacks of availability [47, p.26]. In summary, mashups have a key role in realising an SaaS model in the enterprise based on the architectural concepts of the SOA [47, p.26].

Chapter 3

Enterprise 2.0 Concepts and Trends

“Enterprise 2.0 can’t just be about a wiki here, a blog there forever. Taken together, the emergence and convergence of Web 2.0 and IP communications is what will determine whether there’s truly an Enterprise 2.0. It’s a new architecture defined by easier, faster, and contextual organization of and access to information, expertise, and business contacts whether co-workers, partners, or customers. And all with a degree of personalization sprinkled in.” [24]

Having introduced the origins of Enterprise 2.0 and important, related terms in Chapter two, the present chapter is dedicated to trends in this field. The current section answers the core question, whether Enterprise 2.0 (E2.0) is just another buzzword for a crowd of some collaborative software tools - or more than that. However described, specialised literature (e.g., a number of publications and blog entries) points out, that these concepts not only concern IT-specialists, but also the majority of knowledge workers¹. More precisely researchers of Enterprise 2.0 do not only propose new software packages, but also a reformed business culture. Paying regard to social aspects in state-of-the-art software solutions eases the harnessing of collective intelligence.

Definition In July 2007 Andrew P. McAfee defined Enterprise 2.0 as “the use of emergent social software platforms within companies, or between companies and their partners or customers” [36]. The term itself is motivated by Web 2.0 and stands for technologies and business practices

¹In the present thesis the term knowledge worker is used for employees who mainly use information technology to achieve their goal in the enterprise. Knowledge workers can be considered as potential users of enterprise software solutions.

in a modern, flexible company. McAfee clearly distinguishes between products for individuals and products for companies.

3.1 SLATES and FLATNESSES

SLATES was created by Andrew McAfee, who uses this acronym to “indicate six components of Enterprise 2.0 technologies” [35, p.23ff]:

Search Users must be able to discover the information they are looking for. Although corporate intranets have navigation aids and are maintained by a professional staff, the search experiences are less successful than in the dynamic, uncoordinated Internet.

Links An appropriate link structure, that reflects the opinion of many people, enhances the ability of the search technology. The goal of E2.0 is to boost deep interconnections of enterprise content in the intranets. To accomplish this intention, “let the intranet be built by a large group rather than a small”.

Authoring An ideal Enterprise 2.0 platform supports authoring by granting every worker easy access. “Evidence from Wikipedia shows that group authorship can lead to convergent, high-quality content.”

Tags Let users organize a large amount of content by allowing them to attach simple, one-word descriptions, called tags. Tagging is an efficient way of content categorization, although the created folksonomies² can be redundant.

Extensions Implement algorithms for pattern matching and use the user activity to create “smart” computers. “Amazon’s recommendations were an early example of the use of extensions on the Web.”

Signals Due to the desired user contributions, a constant occurrence of new content is unavoidable. “Even with powerful tools to search and categorize platform content, a user can easily feel overwhelmed”. Therefore there is a need for signals, which notify the users, if new content of interest appears. Novel platforms use technologies like RSS and Atom (see Section 3.4) and JSON (see 3.4) to signal users.

²“a categorization system developed over time by folks”

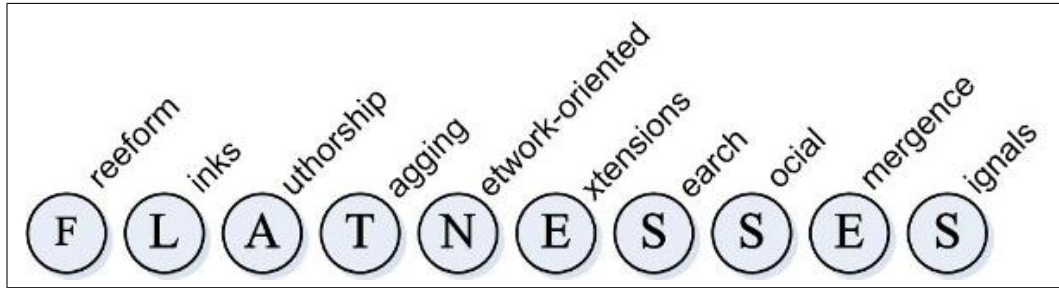


Figure 3.1: FLATNESSES
[21]

In 2007 Dion Hinchcliffe [21] created a more refined conception called FLATNESSES, because he wanted to emphasize four additional properties of Enterprise 2.0. He added the following four words to SLATES, which has resulted in the acronym shown by Figure 3.1:

Freeform Enterprise 2.0 is indifferent to formal organizational identities and there are no barriers to authorship. Andrew McAfee has already emphasised that E2.0 seems to be more freeform than older concepts, like groupware [36].

Network-Oriented The content of an Enterprise 2.0 application must be “fully Web-oriented, addressable, and reusable” [21].

Social

Emergence Emergence means that something complex arises out of relatively simple interactions. Although Andrew McAfee introduced Enterprise 2.0 as the dawn of emergent collaboration [35], the acronym SLATES does not emphasize this characteristic.

3.2 Software Solutions

Software solutions for communication and cooperative work have existed long before the term Enterprise 2.0 has been introduced. *Collaborative authoring tools* (better known as *wikis*) are widely-used knowledge management systems. Experience has shown that not only software developers but also all the knowledge workers are able to use wikis effectively. Another widespread software solution in enterprises are corporate web logs (*corporate blogs*). On the one hand management often considers blogs as modern communication

mediums with one purpose only: to replace static pages with an elegant, dynamic solution. On the other hand comments and feeds (see RSS and Atom in Chapter 3.4) are important features, because they enforce a more democratic and more flexible communication.

To make the content of wikis and blogs searchable, they must be organised. According to SLATES, Enterprise 2.0 products enhance the ability of the search technology with tags and links. As the resources on the Internet or an intranet are managed jointly, the terms *social tagging* and *social bookmarking* are commonly used.

All the tools listed in this section are only expedient if there is a large group of active users. Otherwise blogs would stay empty, wiki content would lose quality, and neither social tagging nor social bookmarking would make sense. Although a standardised definition for Enterprise 2.0 does not exist, every E2.0 specialist (or at least the vast majority of them) would agree that building an online community is crucial. Few years ago it was not easy to make someone's acquaintance in forum and board communities. *Social networking*, however, provides for pervasive linking of content and people [17, p.51].

In summary it can be said that a number of useful software solutions already exist, and many of them are really effective, if the user participation is high. Instead of letting the own IT-specialists combine services and buy different solutions from different vendors, some enterprises prefer a complete platform. These platforms are customized for the client's needs and include features like corporate blogging, collective authoring and even social networking. The next section discusses the relationship of the popular enterprise platform Sharepoint and Enterprise 2.0.

3.2.1 Sharepoint and Enterprise 2.0

On examination of E2.0 software solutions it is necessary to deal with Microsoft Sharepoint. However, first it must be clarified, for what exactly the word "Sharepoint" stands. This section does not introduce the whole history of the platform, but gives a short summary of its products and its relation to Enterprise 2.0. As these products have very much in common, the meaning of the term "SharePoint" is not always unambiguous.

Windows SharePoint Services (WSS) Version 3.0 "is the foundational product that provides a set of building blocks for creating SharePoint applications" [70, p.3]. Although WSS provides the core SharePoint features and services including application management, it is still mainly

used for document management and collaboration [53, p.3] [70, p.3].

Microsoft Office SharePoint Server 2007 “is the server-side infrastructure that turns Office 2007 clients into generators and consumers of content for SharePoint applications. MOSS 2007 is the successor to SharePoint Portal Server 2003” [70, p.3]. Depending on the purchased licence the Standard Edition (MOSS SE) or the Enterprise Edition (MOSS EE) can be installed. “MOSS SE is an application built on top of WSS and it provides the user interface for the basic collaboration/publishing features (...). MOSS EE adds additional enterprise features” [53, p.3].

There are six key functional areas that MOSS 2007 brings together: Collaboration, Portal, Enterprise Search, Enterprise Content Management, Business Process and Forms, and Business Intelligence. The features and services of these areas allow effective collaboration in organisations, which is important for organisations to stay competitive [58].

Microsoft Office SharePoint Server 2007 for Search is also part of the SharePoint Product Line and has both Standard Edition and Enterprise Edition.

“MOSS for Search provides only the SharePoint search capabilities that can be used as a stand-alone search engine and/or be incorporated into other applications. Functionally, there is no difference between the Standard and Enterprise Editions, though the SE edition is limited to a maximum of 500,000 documents.” [53, p.4]

Due to the evolution of Sharepoint Team Services³ and the increased flexibility, WSS/MOSS are widespread in enterprises and are furthermore often considered as Enterprise 2.0 solutions. However, many viewpoints of different E2.0 specialists on Enterprise 2.0 and Sharepoint exist. Dion Hinchcliffe, for example, has listed the issues and challenges of using SharePoint for Enterprise 2.0 [23]:

1. The standard Sharepoint configuration has relatively weak support for the most common Enterprise 2.0 application types.
2. The wilds of the open network (e.g., non-Internet Explorer browsers, mobile devices) can be a challenge for Sharepoint.
3. *“Traditional enterprise systems, including SharePoint, tend to be more rigid in their ability to be shaped by users and too often force users into*

³The original name of the platform, which has been changed to WSS in 2003.

pre-determined uses rather than letting the users shape the use of the tools to best fit the work.” [23]

4. Compared to most Enterprise 2.0 products, complexity and costs are high.

Many of these issues have been resolved by the *Community Kit for SharePoint*⁴ and by *Microsoft Sharepoint 2010*, which is called the “Business Collaboration Platform for the Enterprise and the Web”. Sharepoint 2010 has a new user interface, allows easy customization of a site and has multiple browser support.

Communities and Composites

The capabilities of Microsoft Sharepoint 2010 are grouped into six sections: sites, communities, content, search, insights, composites [41, p.3]. The white papers available for each section make clear, that Microsoft has aimed for resolving the issues above and for converting Sharepoint into a real Enterprise 2.0 platform. Concerning communities it is emphasized, that building corporate communities and social networks is essential [39, p.4]. “A corporate social network builds business communities that cut across departments and geographies” [39, p.7]. In summary the white paper is consistent with the recent research findings. However, the corporation’s terminology differs somewhat from the terminology used in research (and even in the present thesis), which might have marketing reasons. For example, Sharepoint 2010 allows the creation of enterprise mashups, but this capability is described with the word “Composites”. SharePoint Composites empowers “[the] organization to take action with user-driven solutions, increasing user and department satisfaction while allowing IT to focus on high-priority projects and to maintain a stable infrastructure” [40, p.9].

3.2.2 Case Studies

Innovations are the result of a scientific process; they are the realisation of new theoretical approaches. Researchers generally have much practical experience and they examine their theories very closely on practical relevance and usefulness. However, case studies are very important to gain knowledge about the success and the impact of an innovation. In this section, selected Enterprise 2.0 case studies are presented in a nutshell.

⁴“Set of best practices, templates, Web Parts, tools (...) not officially maintained or supported by Microsoft” <http://www.codeplex.com/CKS> .



Figure 3.2: Capability areas of SharePoint 2010
[41, p.3]

Social Networking at Work [26]

The assessment of new technologies is crucial for enterprises. In order to determine the impacts of social networking behind the corporate firewall, a social networking site called Beehive was built and installed in IBM's intranet. Beehive is designed like open social networks: users can create connection, share content and customise content. However, the contact information is fetched from IBM's corporate directory and "this type of linking between intranet services is a unique strength of a tool hosted internally". The concrete numerary of the analysis is described as follows:

"In the year since launching Beehive (May 2007 May 2008), the over 30,000 users on the site had contributed over 250,000 friend connections, 27,000 about-you statements, 36,000 status message updates, 32,000 photos, 10,000 lists, and 100,000 comments on content." [26, p.713]

Overall seventeen users were interviewed, after the tool had been used for several months. The investigation yielded the following results:

Type of connections Interestingly, the primary objective of the employees was not to connect with immediate colleagues, but rather to make new personal and business connections. Beehive helped some users to maintain looser connections within the company. By using the friends-of-friends method and the comment feature, the employees met new colleagues as well. Due to the common ground that exists within the enterprise, the users were more open to new connections, than in an open social network like Facebook.

Reasons of sharing The motivations of sharing content can be classified into three groups. (i) *Caring*: "connecting on the social level was a source of statifaction". (ii) *Climbing*: connecting for career reasons. (iii) *Campaigning*: gathering support for a special cause or project.

Type of shared content What content the employees have shared strongly depends on the main motivation of sharing. Figure 3.3 shows an overview of the shared contents' type.

	<i>Caring</i>	<i>Climbing</i>	<i>Campaigning</i>
Lists	Hobbies, outside interests: <ul style="list-style-type: none"> – Cities I've lived in – 5 Instruments I've played – Favorite American Idol Contestants 	Skills, background: <ul style="list-style-type: none"> – My Practice Areas – 5 IBM Sites I've Worked At – 5 Big Lessons Learned at IBM – My Team (and extended team) 	Thoughts on professional topics and advice: <ul style="list-style-type: none"> – Must See Presentations at [Conference] – What list topics could buzz up a new angle on local growth markets? – Seeking top 5 IT technical skills in demand in AU
Photos	Personal: <ul style="list-style-type: none"> – Family holiday picture – Vacation pictures – The team socializing 	Work-related photos: <ul style="list-style-type: none"> – Headshot – Important conference – Meetings with key people 	Variety of photos: <ul style="list-style-type: none"> – Related to project or campaign
About-you's	Hobbies, outside interests: <ul style="list-style-type: none"> – What do I do when I'm not at work? – Do you call it "pop," "soda," or "coke?" – Where did you grow up? 	Projecting experience, resume: <ul style="list-style-type: none"> – My past projects – Education – What are you most passionate about? – What are your career goals? 	Project description, goals, links to other: <ul style="list-style-type: none"> – Kelly's and my [innovation] idea! – Try the new Web2.0 Intranet Search Engine – What am I up to at the moment?
Status	Activities, state of mind: <ul style="list-style-type: none"> – Is getting another coffee... – Is glad he can play paddle tennis again – On vacation in Tampa Florida! 	Emphasis on current work: <ul style="list-style-type: none"> – Is challenged by time management – Is in Minneapolis. Back Tue July 17th – Is desperately trying to write everything possible out of his last days on assignment in the CIO's office 	Goal and opinion oriented: <ul style="list-style-type: none"> – Is searching for A/NZ based people interested in Web 2.0 tools, knowledge networking, collaboration! – Is busy promoting the E&E video contest (2/25-4/11). Talk it up!

Figure 3.3: Examples of site content, categorized by users goals of caring, climbing, or campaigning

[26]

The main finding of the publication is that both the employees and the companies can benefit from setting up an intranet social network site. Employees can become a part of new communities of practice; furthermore, social network sites can support careers and campaigns. Companies can “bridge generational gaps and boundaries by supporting this method of communication between employees”. If the company does not provide such a system, employees might move outside the firewall and use open social networks like Facebook.

COMET [29]

COMET is a prototype software which was constructed at the Helsinki University of Technology “in order to improve the collaboration and communication between geographically distributed organizations” . The software, which is a phpBB-platform⁵ with integrated RSS-technology, has been used for one year in industrial environments. The case study shows that there are significant differences between company internal and cross-organizational pilot users. “In the company internal COMET the persons knew that every user had approximately same job content and organizational position”, therefore the willingness to post new messages was higher. A critical comment was, that “COMET-like functionality should be more tightly integrated to the existing work systems and practices” .

Wiki Issues - Bosch Diesel and VierD.de

Choosing the right tools for projects of virtual teams is a challenge. Many companies favour the adoption of wikis in order to support collaborative work. It is extremely important not to underestimate the change process; at Bosch Diesel, for example, it took approximately one year. Although wiki systems focus on simplicity, the training of employees was essential. Another “lesson learned” was the importance of templates in case of multiple authorship. The case study of Bosch Diesel has shown that even this kind of Enterprise 2.0 solution requires an appropriate change management [38, p.132 ff]. On the other hand the Second Life project “VierD.de” has shown that using a wiki alone (in that particular case study the teams have worked with Confluence) cannot always satisfy all needs. An issue was the collaborative editing of Powerpoint documents, therefore the team decided to hire a Microsoft Sharepoint Server. Due to the tight integration with Microsoft Office no special training was necessary. Although there were some technical difficulties (e.g., Single

⁵<http://www.phpbb.com/>

Sign-on was not supported), this kind of portal solution with integrated document management system and the integrated wiki was a better alternative, than a wiki alone [38, p.125ff].

Communities of Innovation - SAP and Vodafone

Major companies have their own E2.0 approaches. The German software development and consulting corporation SAP, for example, has started a program called “Communities of Innovation” to let people connect, collaborate and contribute. It has all begun in 2003, when existent and potential customers were invited to discuss about a certain topic: SAP’s NetWeaver. This idea has resulted in an active, successful online community called SAP Developer Network (SDN), which has already nearly one million registered users [71, p.182]. The largest mobile telecommunications network company in the world, Vodafone provides its knowledge workers a number of tools for communication and collaboration, too. The high increase of data volume at the company results in information overload and an emergent need of new approaches. To make such a high amount of data manageable, relevant information must be filtered out. Since relevant information depends on audience, context and time, the filtering mechanism is a real challenge. The majority of Enterprise 2.0 solutions at Vodafone are quite similar to other enterprises. In the first phase a global service for wikis and blogs has been created, while the E2.0-Model itself can be described with keywords such as user participation, collaborative tagging, collaborative authoring, etc. The project “Vodafone Vision” is a social networking approach, which has a main focus on personal knowledge management. The goal is to organize corporate knowledge better by letting the employees create a private area. Users can manage their profile, their contacts, their bookmarks, their pages and their (search)agents [71, p.194 ff].

Cases 2.0

In July 2007 professor Andrew McAfee launched an Enterprise 2.0 repository called Cases 2.0⁶. The goal has been to collect E2.0 case studies based on the provided wiki-template. In the period of two years a total of more than 40 cases have been entered, e.g., *at&t Collaborative Integration*, *Citrix internal blogs*, *Deloitte Southern Africa - Employee Engagement*, *Oracle IdeaFactory* and *Oracle Technology Network*. The solutions described in this

⁶<http://www.cases2.com/>

repository show that many enterprises aim for an online community platform similar to the SAP Developer Network. Oracle Technology Network, for example, is the world's largest community of developers, database administrators, and technical architects, while IdeaFactory helps to share ideas across the entire organization. Many companies consider the use of SharePoint to meet the E2.0 challenges. At&t, for instance, has deployed Microsoft Office SharePoint Server and Windows SharePoint Services in two phases (in 2004 and 2007) and has reported excellent results, including measurable benefits such as increased speed of business and decision making. On the other hand some case studies show that Sharepoint deployment would be too expensive and too long-lasting. Therefore Graymont Intranet is ThoughtFarmer-powered (<http://www.thoughtfarmer.com/>) and Deloitte Southern Africa uses Virtual Works (<http://www.virtualworks.co.za>). Regardless of the used software solution an extensive training is needed, as in Enterprise 2.0 optimally all the knowledge workers should collaborate rather than certain experts.

3.3 Usability Issues

As already mentioned, Enterprise 2.0 is not only about new software packages, but also about a reformed business culture. The majority of knowledge workers is supposed to enhance collaboration and share knowledge by using E2.0 products, i.e. there is a large number of users. Consequently usability is an important consideration in the design of such products. The ISO 9241 standard includes the following definition of usability: "Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [3]. There exist a wide range of publications concerning usability issues. Beaton et al., for instance, have made a usability evaluation for enterprise SOA APIs [6]. However, the most cited resources are the works of Jakob Nielsen written in the early nineties: the paper "Heuristic evaluation of user interfaces" [45] and the book "Usability Engineering" [44]. If a new tool was not conform to the usability requirements, users would lose motivation and would prefer to use applications they are familiar with (e.g., email, instant messaging). However, the objective of Enterprise 2.0 is a comprehensive adoption of emergent, freeform, social software. Users desire straightforward, light-weight tools that do not overwhelm them. The following sections examine usability issues in the E2.0 context and propose solutions where it is necessary.

Installation

The appropriate installation of software is crucial for enterprises. Furthermore configuration and updating are supposed to work reliable - independent of the complexity. The majority of software houses has already tried to optimize these processes by enhancing usability. However, a few issues still exist:

- setup and configuration are too complex, adjusting the software to users' individual needs is not possible or is too difficult
- updates are pushed too frequently, do not work automatically or/and constrain the users in their work
- reinstallation of workstation means reinstallation of all the software

Social software solves some of these problems (e.g., task appropriateness and individualization), since it is often constructed very modular and adaptable. Users can select the components that are relevant to their tasks and do not need to put up with additional functionality [38, p.119]. In addition, social software is a solution for the basic installation problems, because it is mostly web-based and therefore rapidly and easy available. It is certainly useful to relocate business software into the Internet as a service. Although web-based business software shows a rising trend, the “Web-based Operating System” [62] - abbreviated WebOS - seem to be utopia these days. In respect of installation and configuration it is advantageous to run all the applications on the top of a browser (which is the WebOS Vision), but it is not yet a realistic concept, due to security issues etc. On the other hand new approaches like AJAX have made it possible to create web applications, that have a level of interactivity (and usability) like desktop applications. Many software companies have promising WebOS approaches, which will be discussed in Chapter 5.

User Participation

The adoption of social software is not only about the selection and configuration of a technical product. In addition, the type of use and the necessary general conditions should be clarified [38, p.131]. The clarification of responsibilities and benefits of each user is reasonable and very important for a successful Enterprise 2.0 project. One reason why E2.0 projects fail is a “lack of effective participants [which results in] empty blogs, wikis, or silent social networks” [22]. The first step to boost participation is to involve users already in the requirements analysis. A good foundation of a user centered design are

user study methodologies, like interviews, contextual inquiry and working with focus groups. Later in the software development process it is recommended to carry out usability testing with real users. Some businesses might find these methodologies too expensive or too tedious; however, long-term experiences do not confirm that opinion. Whatever solution a company goes for, the objective must be the enabling of real user participation rather than the application of some methodologies as pretences.

Consistency and Standards

Based on sections 3.1 and 3.2.1 Enterprise 2.0 products should have no barriers to authorship, more precisely these products should be flexible in their ability to be shaped by users. On the other hand consistency and “interface standards lead to ease of learning and ease of use” [44, p.227], therefore improve users’ productivity, which is also an objective of E2.0.

In 1993 Jakob Nielsen has not only written about the benefits of consistency and standards, but also about the danger of them. “The very idea of consistency also implies reduced flexibility in the design of individual products so they may not be able to be as tailored to application-specific requirements or context” [44, p.230]. In summary user interfaces should be both consistent and flexible:

- users should rarely or never face unexpected situations
- the emergent collaboration (see Section 3.1) should be possible

3.4 Underlying Technologies

Experts of Enterprise 2.0 or any other IT initiative often discuss whether the concrete technology is an important factor or not. Many vendors and consultants tend to encourage the view that, for example, there is only a need to collaborate and share knowledge better, but it’s not about the technology. On the one hand the success of modern approaches strongly rely on social factors and a piece of technology alone will not generate benefits. On the other hand managers must not underestimate change management technologies can not only differ from each other in salient ways, but they can change over time [71] [37]. The exhaustive discussion of this topic goes beyond the scope of this work. The present section describes fundamental technologies for implementing social software (or Web 2.0 applications in general) and OpenSocial

applications.

Representational State Transfer (REST)

REST originated from Roy Fielding’s doctoral dissertation and describes a software model for effective interactions and data delivery in a distributed hypermedia system [33, p.169].

“The name “Representational State Transfer” is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through the application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use.” [14, p.109]

REST was derived as an architectural style by applying the constraints *client-server*, *stateless*, *cacheable*, *uniform interface*, *layered system* and the optional constraint *code-on-demand* [14, p.76ff]. In the majority of cases the software designed in conformance with this style uses HTTP, more specifically the HTTP methods (e.g., POST, GET, PUT or DELETE). Web-based RESTful systems⁷ mainly establish a mapping between HTTP methods and read, update, and delete (CRUD) operations [50]. POST is used for creating a resource on the server, GET is used for retrieving a resource, PUT is used for changing the state of a resource or for updating the resource, and DELETE is used for removing a resource [50]. Figure 3.4 shows how a GET request can be turned into a RESTful POST request, and the example in Figure 3.5 shows how HTTP PUT can be used to update a user.

RSS and Atom

Due to the characteristics of Web 2.0 applications discussed in Section 2.4 there is an increased need for publishing new content in a structured way. For example, users subscribe to web feeds of frequently updated blogs, more precisely register with a feed reader. Typically this kind of software is able to read the widespread Really Simple Syndication (RSS) and Atom. *Feeds* are “composed of a number of items, known as “entries”, each with an extensible set of attached metadata” [46, p.2]. Both the proposed standard of Atom [46] and

⁷Systems conforming to the REST.

HTTP GET request

```
GET /adduser?name=Robert HTTP/1.1
```

↓

HTTP POST request

```
POST /users HTTP/1.1
Host: myserver
Content-Type: application/xml
<?xml version="1.0"?>
<user>
<name>Robert</name>
</user>
```

Figure 3.4: REST Example 1: creating a user with HTTP POST
[50, p.3]

Update over HTTP GET

```
GET /updateuser?name=Robert&newname=Bob HTTP/1.1
```

↓

HTTP PUT request

```
PUT /users/Robert HTTP/1.1
Host: myserver
Content-Type: application/xml
<?xml version="1.0"?>
<user>
<name>Bob</name>
</user>
```

Figure 3.5: REST Example 2: update a user with HTTP PUT
[50, p.4]

the RSS 2.0 Specification [72] show, that these web syndication formats are XML-based. In practice they support collaboration and communication very well, therefore they are used in many software systems, e.g., Microsoft's SharePoint. Besides RSS and Atom the JavaScript Object Notion (see subsection below) is a widely used format.

JSON

JavaScript Object Notation (JSON) is a data-interchange format, which is not only easy to parse for machines, but easy to read for humans. JSON can be considered as an alternative to XML-based data formats. It “is a good fit with OpenSocial because it represents data with JavaScript literals (...) There is the added advantage that the overhead of JSON is less than XML” [33, p.5]. In Figure 3.6 there is a valid JSON example that contains an object, an array, some strings and a number. All these types are shown in Figure 3.7⁸. For more details on JSON structures see Appendix B.

```
{
  "name": "OpenSocial RESTful Protocol",
  "version": "0.8.1",
  "formats": [
    "RSS",
    "Atom",
    "JSON"
  ],
  "year": 2008
}
```

Figure 3.6: Example of a valid JSON

```
object {
  "name": string "OpenSocial RESTful Protocol",
  "version": string "0.8.1",
  "formats": array [
    string "RSS",
    string "Atom",
    string "JSON"
  ],
  "year": number 2008
}
```

Figure 3.7: Example of a valid JSON completed with types

⁸Types are just annotations and not part of JSON.

Asynchronous JavaScript and XML

In context of social software implementation the word AJAX, the acronym for asynchronous JavaScript and XML, is frequently used. It is not a technology in itself, but a term that embraces new approaches for the development of Web 2.0 applications. AJAX applications have six typical attributes based on [17, p.671-672]:

- User interfaces getting more dynamic
- First step to Rich Internet Applications
- Individualisation based on user interaction
- Redundancy for maximal coverage
- Robust communication of the application (progress indicators,etc..)
- Web applications are alike desktop applications

AJAX is a combination of well-established technologies like *XHTML*, *CSS*, *DOM*, *XML* and *JavaScript* completed by the so-called *XMLHttpRequest* object. The *XMLHttpRequest* object assumes the core task of generating and handling clients' requests asynchronously during the interaction of the user [17, p.676]. Although the history of XHR has already begun around the turn of the millennium and all the popular browsers have fully implemented it, the last revision of the World Wide Web Consortium to the specification in 2008 was still a working draft [60].

JavaScript Libraries and Web Application Frameworks

Technologies, that allow the development of AJAX applications, are usually encapsulated in JavaScript libraries and web application frameworks. JavaScript libraries (e.g., *Dojo Toolkit*, *Prototype*, *Script.aculo.us* or *jQuery*) include features like DOM traversal and modification, CSS manipulation, effects and animations. Web application frameworks “aim to alleviate the overhead associated with common activities performed in Web development” [69]. Many languages - for example Perl, Python, PHP, Java, Ruby - have an associated web application framework [69]. A popular framework is *Ruby on Rails*, which integrates *Prototype* and *Script.aculo.us* and supports REST as well. Microsoft developed ASP.NET, which is the successor of the Active Server Pages (ASP) technology. “The release of ASP.NET 2.0 [in 2005] provided the foundational

layer that was missing in previous versions of SharePoint” [70, p.2]. ASP.NET 3.5 - release date November 19, 2007 - includes both ASP.NET AJAX and WCF (Windows Communication Foundation) support for RSS and JSON.

3.5 The Influence of CSCW and Web 2.0

“CSCW should incorporate successful Web 2.0 patterns in addition to the existing Groupware and focus on making the ideas and tools from the two fields work together.” [28, p.425]

Having discussed Enterprise 2.0 in detail, the present section aims for clarifying the influence of CSCW and Web 2.0. More precisely, the relationship to these terms must be clarified. Enterprise 2.0 is, reminiscent of CSCW, an interdisciplinary research field: findings of sociology, psychology, anthropology, work science are all relevant. Enhancing efficiency and flexibility of cooperative work, for example, is impossible without analysing the tasks based on CSCW-relevant aspects. By using the ethnographic methods of sociology the employees and their work can be observed, and models of behaviour can be documented. The objective of CSCW is to understand collaboration and to create socio-technical systems for supporting collaboration [28]. The term groupware, which is used for “technologies and tools that facilitate shaping these socio-technical systems” [28, p.419], has been introduced in Section 2.1. Besides the interaction of humans, artifacts and operations are a central point of both CSCW and Enterprise 2.0 research. Artifacts are shared, therefore they must be accessible for every member of a group, and group members operate on these common artifacts. Common artifacts are characterised by a predictable structure and function, and they allow the communication within the team [55]. Common artifacts offer an overview of cooperative work done by the actors [55], and signals (see SLATES in Section 3.1) are supposed to notify the actors about changes.

A group of interacting human individuals are called a *community* as well. The exact definitions of the term “group” and the term “community” are subjects of a sociological debate, but such a discussion goes beyond the scope of the present work. Section 2.4, however, has clarified some core definitions related to *online communities*, because these concepts have influenced Enterprise 2.0 very much. Web 2.0 is characterised by active user participation, a high amount of user generated content and arising online communities. The advantage of working

	Groupware	Social Software
<i>communication</i>	group oriented(“we”)	person/self oriented(“me”)
<i>implementation</i>	top down	bottom up
<i>participation</i>	enforced	voluntary
<i>cooperation</i>	pre-planned ways	co-evolved conventions
<i>number of users</i>	small	large
<i>period of time</i>	limited	no project limitations

Figure 3.8: Differences between groupware and social software [28]

in such a community is the wisdom of crowds, which is described with the term *network effect* in economics [17, p.43].

Network effect If new users participate in creating a product in a community, both the value of the product will enhance and all the participators will benefit [17, p.43].

As a result of the network effect, harnessing collective intelligence is also characteristic for Web 2.0 [48]. Consequently, there are many commonalities between social software (a subclass of Web 2.0) and knowledge management. In actual fact, both groupware and social software are considered as software solutions for knowledge management (compare Section 2.3). Although Michael Koch listed some differences between groupware and social software (see Figure 3.8), he also introduced a integrated perspective of CSCW and Enterprise 2.0 [28].

Regarding knowledge management, *communities of practice* and *communities of interest* are widely discussed, and the research related to this topic is an important part of the theoretical fundament of Enterprise 2.0.

Communities of practice are “groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis” [63, p.4]. The relationships within the community are not organisationally determined [31, p.832].

Communities of interest “bring together stakeholders from different CoPs to solve a particular (design) problem of common concern” [15, p.4].

Communities of practice can be considered as homogenous communities, while communities of interest are heterogeneous [15]. Referring to E.L. Lesser and J.

Storck, communities of practice create social capital, and the existence of social capital improves an organisation's performance [31, p.833]. Enterprise 2.0 concepts and the underlying technologies support the formation of communities of practice, therefore facilitate the emergence of social capital. Furthermore Enterprise 2.0 approaches are usually in accord to the suggestions of Lesser and Storck:

- “Provide opportunities for individuals to make new connections.” [31, p.840]
- “Allow time and space for relationship building among individuals.” [31, p.840]
- “Find ways to communicate the norms, culture, and language of the community and the organization.” [31, p.840]

3.6 Security and Privacy

The present chapter has introduced the definition of Enterprise 2.0 and has analysed concepts, trends and software solutions. Furthermore the usability issues, the underlying technologies and the influence of CSCW and Web 2.0 have been discussed. To complete the description of Enterprise 2.0, it is unavoidable to examine security and privacy issues as well. Both developers and users have to be aware, that modern network communication brings not only opportunities but also risks to an enterprise. Before these risks can be identified, the context, the communication etc. has to be analysed. By using a very simple categorisation, Enterprise 2.0 interactions can be classified into following three groups: human-to-human interactions, human to computer interactions and communication between computers. Consequently threats can originate from two major “sources”: human individuals (or groups) and computers (or more precisely technology). Having a secure technological background (e.g., firewalls, secure programming) is a required but not a sufficient factor. It is very important to deal with human aspects as well. Even articles in non-specialised, daily newspapers report on privacy protection in social networks and in the Web in general, since many users are affected by technical achievements. Therefore Section 6.4 lists possible threats and calls for a responsible use of new technologies.

Chapter 4

OpenSocial™ in Action

A core question of the present thesis is whether or not OpenSocial can be applied successfully in Enterprise 2.0 projects; therefore the present chapter describes this particular standard. First, the most important facts of the specification are introduced and the basics of application development are explained based on two simple examples. Second, the deployment in OpenSocial containers is discussed and popular sandboxes are evaluated. Finally, the application implemented within the scope of the present thesis is documented.

OpenSocial is a set of open application programming interfaces (APIs) for social network development. It is “designed to ease deployment of a single application to multiple social network” [33, p.55]. The non-profit OpenSocial Foundation “helps facilitate the development of new specifications and ensures that the technical direction remains in the hands of the community”¹. As of the writing of this work the following versions have been released:

- **v0.7** “considered by many to be the first truly functional version” [33, p.56], released on January 25, 2008
- **v0.8** supported by all the major social network platforms, released on May 27, 2008
- **v0.8.1** released on September 25, 2008
- **v0.9** supported by MySpace and RenRen (China), released on April 15, 2009
- **1.0** specification published on March 9, 2010

¹OpenSocial Foundation FAQ, last visited on 16th February 2010
<http://www.opensocial.org/page/opensocial-foundation-faq>

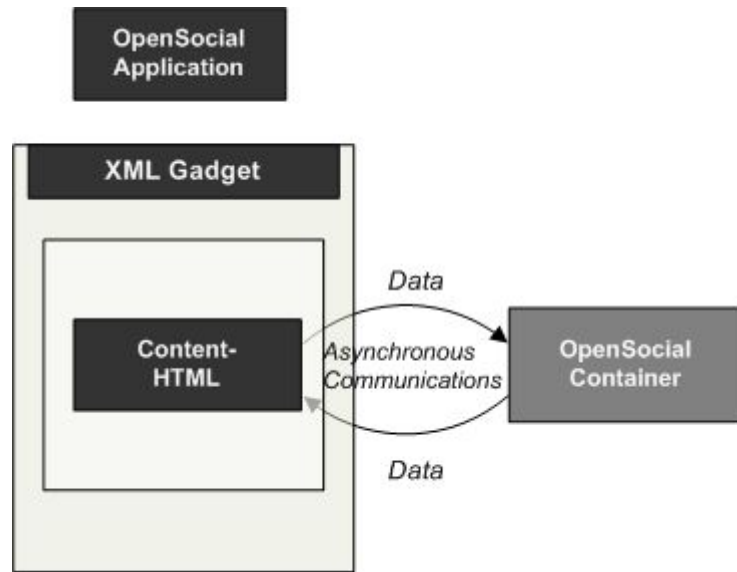


Figure 4.1: Application architecture
[33, p.63]

With version 0.8, there are two OpenSocial application programming interfaces: the JavaScript API and the server-based RESTful API. The RESTful protocol “enables developers to write applications that are hosted on a server and make direct calls to the container” [33, p.58]. The JavaScript API is used by client-based OpenSocial applications and can implement application-specific functionality, Gadget API calls and OpenSocial API calls [33, p.63]. It is recommended to read the specification and the official API reference to get a profound knowledge of OpenSocial. The specification includes, inter alia, the section “Compliance”, which lists and explains the requirements of an OpenSocial container. A reference implementation of the specification is Apache Shindig.

4.1 Application Development

“An OpenSocial application is, at heart, a combination of XML and JavaScript, using a special version of Google Gadgets. The code is written in JavaScript, and preferences and guidelines [...] are set using XML” [30]. Figure 4.1 shows the architecture of such an application.

Consequently the first step of OpenSocial network programming is to become familiar with Google Gadgets. Just like in Listing 4.1, a basic Google Gadget is an XML file that contains a `<Module>` element. Within the `<Module>`

element there are three potential sections: *ModulePrefs*, *UserPrefs*, *Content*.

ModulePrefs “defines the settings for a particular gadget” [30]. Inside *ModulePrefs* there are both attributes and elements. The example in Listing 4.1 contains the attributes *title*, *author*, *author_email* and two *Require* elements.

UserPrefs is used to set user preferences [30].

Content combines the gadget attributes and user preferences with programming logic and formatting information to become a running gadget². In other words the content section can contain HTML, CSS and Javascript, or references to external resources.

The basic Google Gadget in the example below sets the gadget’s height and title based on the value of the UserPref “height”. Before the special functions `gadgets.window.adjustHeight()` and `gadgets.window.setTitle()` can be used, the feature dependencies “dynamic-height” and “settitle” have to be declared. This is done in the “Require” tags (line 5 and line 6). In addition the exemplary gadget contains a minimal CSS and a minimal HTML section.

²Gadgets XML Reference, last visited on 16th February 2010
<http://code.google.com/intl/en-EN/apis/gadgets/docs/reference.html>

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Module>
3     <ModulePrefs title="Basic Google Gadget" author="Oliver Grof"
4     author_email="e0227374@student.tuwien.ac.at">
5         <Require feature="dynamic-height" />
6         <Require feature="settitle" />
7     </ModulePrefs>
8     <UserPref name="height" default_value="200" required="true" />
9     <Content type="html">
10        <![CDATA[
11            <script type="text/javascript">
12                var prefs = new gadgets.Prefs();
13                var height=prefs.getInt("height");
14                gadgets.window.adjustHeight(height)
15
16                var myTitle="Basic Gadget - "+height+"pixels high";
17                gadgets.window.setTitle(myTitle);
18            </script>
19            <style type="text/css">
20                #content {color:blue}
21            </style>
22            <div id="content">Some Content</div>
23        ]]>
24    </Content>
25 </Module>

```

Listing 4.1: A basic Google Gadget created for this thesis

To see a gadget in action, it has to be added to iGoogle (<http://google.com/ig>). Due to the fact that iGoogle is, like Documents, Groups, etc., one of the numerous services provided by Google, the creation of a personal page requires the registration of a Google account. Another requirement is that the gadget has been published, either on a private Web Server or via the Google Gadget Editor (GGE). GGE “*not only lets people edit their own gadgets via a Web browser, but also provides free storage for gadgets*” [30]. The adding itself can be completed by clicking on the link “Add stuff” and then on “Add feed or gadget”.

By using the OpenSocial API, however, gadgets can be turned into social applications. A few specific features of a basic OpenSocial application are treated in Listing 4.2. First, the gadget has to require the opensocial feature, where the version depends on the container. In this particular example, the version

0.8 is used (see line 6). Second, the JavaScript code for communication has to be implemented. “Most of the OpenSocial methods do not directly return data. Instead, the application must request this data, and [...] the container will return the results to a callback function” [33, p.63]. The application in Listing 4.2 requests two Person objects and calls `getDiplayName` for both objects in the callback function named `Application.displayContent`. In detail: The three kinds of people in OpenSocial are *owner*, *viewer* and *friends*, all of them are represented by a Person object. In the example below, a new request is created in `init()` to get the owner (the user, who has installed the application) and the current viewer of the application. Next, in line 25, the request is sent to the container, with the callback function as a parameter. Finally, the HTML is built added to the “main” element. Note: The function `_gel()` is a wrapper around `getElementById()`.

In addition to **people and relationships** there are two feature areas, which are not discussed in Listing 4.2:

Persistence The OpenSocial API allows to save and load data for each of the application’s users. Data is stored as a series of key/value pairs.

Activities The actions performed by the user can be shared on the activity stream. “An activity could be anything from modifying an application’s state to writing an online review for a movie”³.

Due to the fact that in some cases it is more meaningful to store data in an external database, the present thesis does not deal with the Persistence API in detail. However, the activity stream is a subject of discussion.

³Gadget Developer’s Guide (v0.8), last visited on 16th February 2010
[http://wiki.opensocial.org/index.php?title=Gadget_Developer's_Guide_\(v0.8\)](http://wiki.opensocial.org/index.php?title=Gadget_Developer's_Guide_(v0.8))

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Module>
3     <ModulePrefs title="PersonRequest" author="Oliver Grof"
4     author_email="e0227374@student.tuwien.ac.at"
5     description="Requesting VIEWER and OWNER">
6     <Require feature="opensocial-0.8"/>
7     <Require feature="dynamic-height"/>
8     <Require feature="settitle"/>
9     </ModulePrefs>
10    <Content type="html">
11    <![CDATA[
12    <script type="text/javascript">
13    var Application = {
14    init:function(){
15        var dataRequest = opensocial.newDataRequest();
16
17        var ownerRequest = dataRequest.
18        newFetchPersonRequest(opensocial.IdSpec.PersonId.OWNER);
19        var viewerRequest = dataRequest.
20        newFetchPersonRequest(opensocial.IdSpec.PersonId.VIEWER);
21
22        dataRequest.add(ownerRequest,'owner');
23        dataRequest.add(viewerRequest,'viewer');
24
25        dataRequest.send(Application.displayContent);
26    },
27    displayContent:function(data){
28        var owner = data.get('owner').getData();
29        var viewer = data.get('viewer').getData();
30
31        var html = 'Welcome '+viewer.getDisplayName() + '!<br/>';
32        html = html+'The owner\'s name is: '+owner.getDisplayName();
33
34        _gel('main').innerHTML = html;
35    }//end displayContent
36    }//end Application
37    </script>
38    <div id='main'></div>
39    <hr>
40    <script>Invitation.init();</script>
41    ]]>
42    </Content>
43 </Module>

```

Listing 4.2: A basic OpenSocial application created for this thesis

4.2 Deployment in Sandboxes

OpenSocial applications run on containers that fulfil the requirements of the OpenSocial specification. A high number of social network platforms are such a container, therefore they are able to host social applications. In the most common situation, the gadget XML file and the associated resources need to be published (e.g., on a private Web server) before they can be deployed. Developers typically do not test their applications in the production environment, but rather in sandboxes. The present section describes the sandboxes and the developer portals of four popular social network platforms, and summarises the evaluation results.

iWiW

iWiW (abbreviation for International Who is Who) is a very popular social network platform in Hungary started in April 2002. In December 2008 the developer portal has been released to support OpenSocial network programming. Developers need both a valid email address and a mobile number of a Hungarian mobile network operator for the registration, because the activation code is sent in an SMS text message. Once the registration is finished, the users can log in and manage their applications. The development is supported by a developers' blog and a wiki; both of them can be read without having logged in, actually. iWiW provides a very advantageous and autonomous sandbox, where the own applications can be tested. The developers' wiki⁴ describes the sandbox as follows:

- To protect one's ideas, everyone sees only his or her own applications.
- Besides the developer's own identity 300 additional users are created, all of them having a number of relations and the same password as the developer. The email addresses are derived from the developer's address. If, for example, it is `oliver@mydomain.com`, the generated users will have addresses like `oliver@t1.mydomain.com`, `oliver@t2.mydomain.com`, `oliver@t3.mydomain.com` and so on.
- The sandbox contains almost every functionality of the production environment, but is a safe 'playground' for developers.

⁴iWiW Homokozó (english: The iWiW sandbox), last visited on 17th February 2010
http://dev.iwiw.hu/wiki/index.php/iWiW_Homokozó

Due to the fact that the main target group of iWiW are Hungarian users, the social network service's infrastructure is much simpler than the international competitors' infrastructure. Therefore iWiW is only available in Hungarian and the sandbox is frequently unavailable due to maintenance.

orkut

Orkut is a social network platform owned by Google, therefore users can log in with their Google Account to use this service. It was the first social network, that supported OpenSocial, even the versions lower than 0.7. The sandbox of orkut is not an autonomous system like the iWiW sandbox. In fact, the only difference to the production environment is the possibility to add own applications. It is recommended to create some dummy users for testing purposes, because otherwise developers have to test their applications with real friends. Orkut allows the use of the server-to-server protocol REST; however, only a subset of the OpenSocial REST API is supported currently.

hi5

Users, who have a hi5 account, can register as a developer at the *hi5 developer platform*. The hi5 sandbox is not a testing environment, but an information center. Developers can find a full SOAP API, and even a few REST endpoints there. In hi5, testing an application with dummy users is more complicated than in other social networks because the hi5 email must match the `author_email` in the application's XML `ModulePrefs`.

MySpace

The MySpace Developer Platform enables developers to create OpenSocial applications. In MySpace, unlike in other platforms, developers must specify a unique email for each application, since a separate application account is created [33, p.68]. The new application accounts are automatically listed as friends of the developer. Uploading the gadget XML, editing the applications information and editing the application source are the next possibilities to continue with the deployment process. The built-in gadget editor allows the developers to edit the canvas, profile and home surface separately and even to save changes in the source.

Evaluation Results

At present, portability is a big issue because the particular platforms support different versions of the OpenSocial specification and have furthermore different restrictions on the specification. It is recommended to read the API documentation of each platform, which will host the application. Within the scope of the present thesis, however, it is not necessary to operate on multiple platforms, but to have a robust, stable sandbox. For the purposes of the present work orkut seems to be the most appropriate system. Both creating dummy users and adding a new application is easy and fast in orkut, in addition it is available in English, which is quite useful, when doing some user interviews outside Hungary.

4.3 Social Conference Manager

To stay competitive, enterprises must invest both in research and development, and in further education. Therefore some knowledge workers have to participate in conferences, where new findings are presented and tutorials, important meetings are held. Within a particular community (e.g., employees of a company) the participation must be coordinated, more precisely it has to be clarified who attends which conference et cetera. In addition, the attendees could socialize during the preparation and even communities of practice could arise, if the users had the right tool for that. The users of a social network platform, for example, could subscribe to a conference and could exchange opinions and information. For this reason, an application should be implemented based on the OpenSocial API, which is suitable for these tasks. The application's use cases in 4.3.1 are based on the following feature list:

- The users can easily install the application.
- A welcome page, that contains updates and other useful information, appears first.
- Available conferences are displayed in application.
- The users can see conferences' details, like date, place, etc.
- Every user can subscribe to/unsubscribe from a conference group.
- All the users, who have subscribed to a conference are displayed.

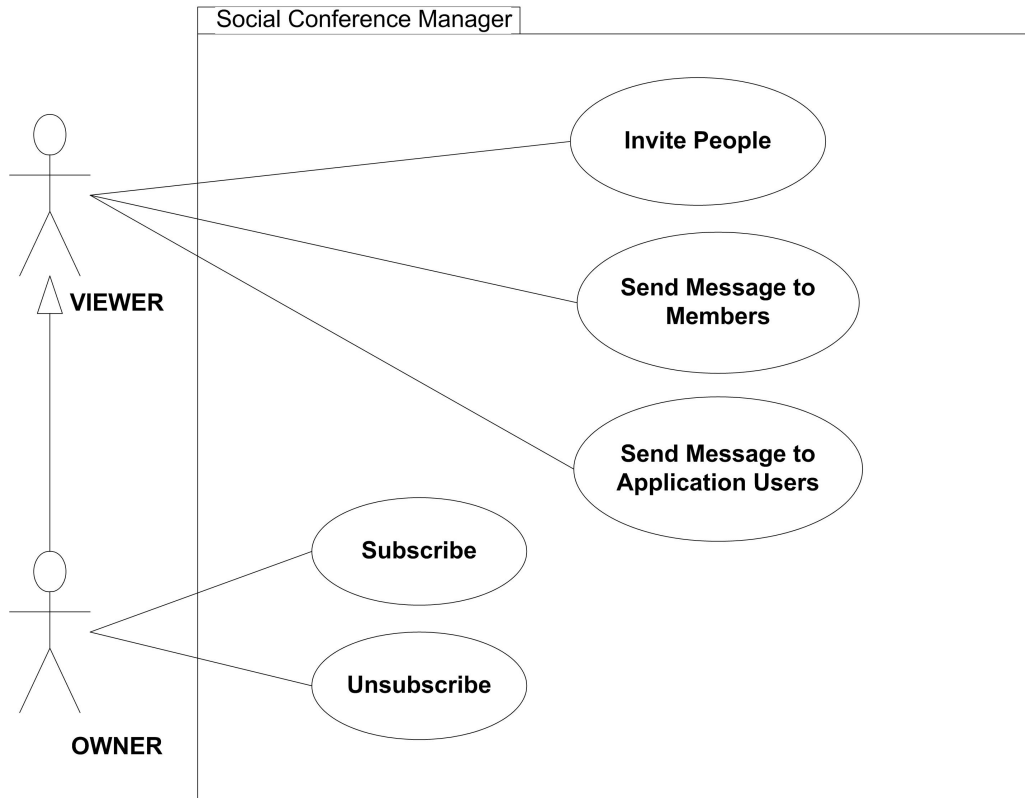


Figure 4.2: Use case diagram of the Social Conference Manager

- Messages can be sent to both group members and all application users.

The basic course of the application's installation depends on the OpenSocial container and is therefore not described in any use case. All the other features are integrated in the use cases.

4.3.1 Use Cases

The present section contains the textual description of the use cases. The textual description adhere to a simple schema with seven obligatory fields and one optional field: *Name*, *Description*, *Actors*, *Triggers*, *Precondition*, *Postcondition*, *Basic course of events* and *Error states*.

Subscribe User

Name	Subscribe User
Description	Users can subscribe to a chosen conference.
Actor	User (OWNER)
Trigger	User wants to subscribe to a conference. (=User wants to be a member of the group that is interested in this conference.)
Precondition	User has opened his or her own application. (= User is the OWNER.)
Postcondition	User is subscribed to the conference. User is shown as a member of the corresponding group.
Basic course of events	<ol style="list-style-type: none">1. User chooses a conference.2. System shows the conference data and the subscribed users.3. User clicks on the button or link “Subscribe”.4. System asks for confirmation.5. User confirms his or her intention.6. System updates the members’ list, System changes “Subscribe” to “Unsubscribe”, System adds the link “Send Message to Members”.
Error states	<ul style="list-style-type: none">• User is not subscribed to the conference.• User Interface is not updated.

Unsubscribe User

Name	Unsubscribe User
Description	Users can unsubscribe from a chosen conference.
Actors	User (OWNER)
Triggers	User wants to unsubscribe from a conference. (=User does not want to be a member of the group that is interested in this conference.)
Precondition	User is subscribed to the conference
Postcondition	User is not subscribed to the conference. User is not shown as a member of the corresponding group.
Basic course of events	<ol style="list-style-type: none">1. User chooses a conference.2. System shows the conference data and the subscribed users.3. User clicks on the button or link “Unsubscribe”.4. System asks for confirmation.5. User confirms his or her intention.6. System updates the members’ list, System changes “Unsubscribe” to “Subscribe”, System removes the link “Send Message to Members”.
Error states	<ul style="list-style-type: none">• User is not unsubscribed from the conference.• User Interface is not updated.

Send Message to Members

Name	Send Message to Members
Description	User can send messages to other users, who are subscribed to a conference (i.e. are members of the group that is interested in a certain conference).
Actors	User (OWNER or VIEWER)
Triggers	User wants to send a message to a certain conference.
Precondition	User is subscribed to the conference.
Postcondition	Members of the chosen conference have all received a message.
Basic course of events	<ol style="list-style-type: none">1. User chooses a conference.2. User clicks on the button or on the link 'Send Message to Members'.3. System opens the form that includes the textarea for the message.4. User enters his or her message and clicks on button or link 'Send'.5. System sends the message to all members (including the sender).
Error states	<ul style="list-style-type: none">• Message is not sent to the members.

Send Message to Application Users

Name	Send Message to Application Users
Description	Users can send a message to all the friends, who have added the application.
Actors	User (OWNER or VIEWER)
Triggers	User wants to send a message to all the friends, who have added the application.
Precondition	The application SCM has been added.
Postcondition	All the friends, who have added the application, have received a message.
Basic course of events	<ol style="list-style-type: none">1. User chooses the welcome page (first tab).2. User clicks on the button or on the link 'Send message to all'.3. System opens the form that includes the textarea for the message.4. User enters his or her message and clicks on button or link 'Send'.5. System sends the message to all friends, who have added the application (including the sender).
Error states	<ul style="list-style-type: none">• Message is not sent to the members.

Invite People

Name	Invite People
Description	Users can send an invitation their friends (=Users can invite one or more friends to use the application).
Actors	User (OWNER or VIEWER)
Triggers	User wants to invite some friends.
Precondition	User has opened the application at least as a VIEWER.
Postcondition	Invitations have been send.
Basic course of events	<ol style="list-style-type: none">1. User selects one ore more friends.2. User clicks on the button or the link 'Invite'.3. Systems sends invitations.
Error states	<ul style="list-style-type: none">• Invitations are not sent.

4.3.2 Implementation Details

The Social Conference Manager has been implemented in the Aptana Studio 2.0⁵, which is freely available both as an Eclipse plug-in and as a stand-alone integrated development environment (IDE). The present section documents important details of the application. First, the technologies used are listed and the architecture is explained. Second, the implemented features are described.

Technologies and Architecture

Having evaluated the social network services iWiW, orkut, hi5 and MySpace, I had to decide whether or not to develop on multiple platforms, and if not, which platform is the most adequate. As already mentioned in Section 4.2, the most appropriate social network platform for the present thesis' objectives is orkut. Regardless of the container, the Social Conference Manager stores the conferences' data and the registrations in an external database. Therefore the application is client-based with application server support. In concrete terms, the following technologies have been applied:

Container	prod.sandbox.orkut.com
PHP Version	5.1.6
MYSQL Server	5.0.45
Apache Server	2.2.3
jQuery	1.4.2
jQuery UI	1.8rc3 (Core, Widget, Mouse, Position, Draggable, Resizable, Accordion, Dialog; Sunny Theme)
PHP OAuth Lib	Revision 1171 from http://code.google.com/p/oauth/

As Figure 4.3 shows, each request to the application server is a signed request. By using the oAuth protocol, the application server acts as a service provider that only allows access for authorized consumers. In the present solution the OpenSocial containers iGoogle, iwiw, hi5 and orkut are such a consumers, because in the design phase it was assumed that the application will run on multiple platforms. Unlike MySpace, the containers named above have all unique public certificates, therefore the server can identify which consumer the request has sent. Due to the fact that security is a critical issue, the intention was to show how the security of a server-supported OpenSocial application can enhanced by using an authentication protocol. Surely, it would be desirable to use private certificates, but this is not relevant for the present project.

⁵<http://www.aptana.org/>

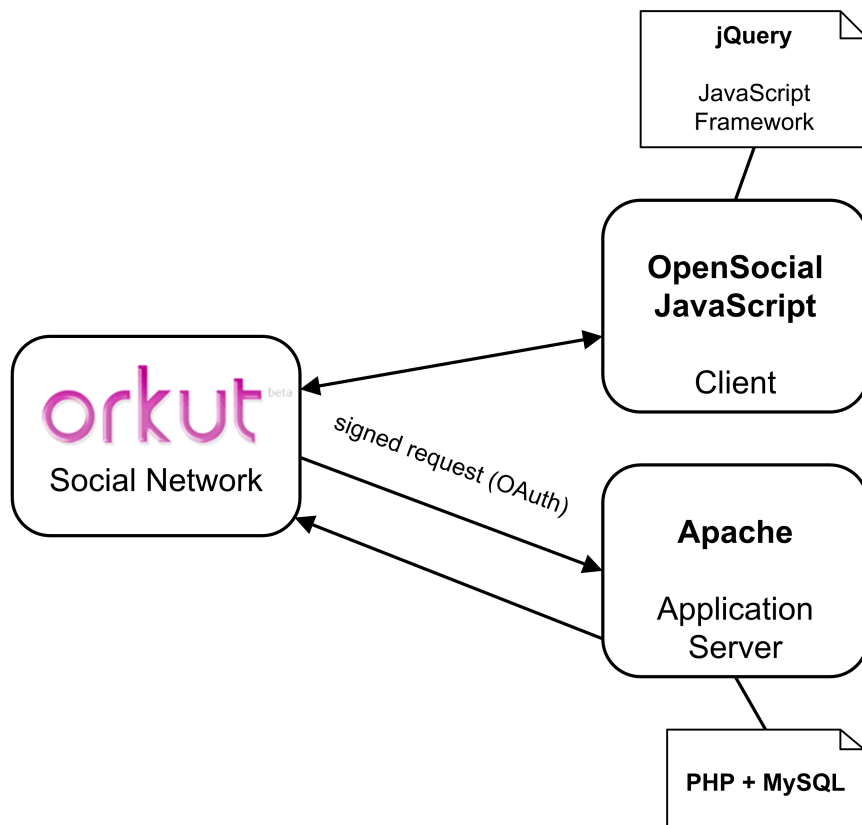


Figure 4.3: Application architecture (orkut)

event	
<i>id</i>	int(10)
name	varchar(250)
date	date
place	varchar(50)
container	varchar(20)

user2event	
<i>userid</i>	varchar(50)
<i>eventid</i>	int(10)

Figure 4.4: Database structure

Note Every application using the OpenSocial REST API as a server to server protocol, is forced to send a signed request to the social network. In this case, the application server acts as a service consumer, which uses OAuth to get access to the resources on the container (the service provider).

The database on the application server consists of two tables (see Figure 4.4): the table **event** stores the information of the conferences and the table **user2event** stores the registrations of the users. In the current version of the application the events are container-specific. Due to the the authenticating mechanism the client only receives the conferences of the hosting container. It should be noted that the server responses contain XML-formatted data.

The application has been implemented in Aptana Studio 2.0⁶, which is freely available both as an Eclipse plug-in and as a stand-alone integrated development environment (IDE).

Implemented Features

OpenSocial applications should be characterized by simplicity and a straightforward user interface. In businesses, there is need for enterprise mashups that are consuming certain services and have “a degree of personalization sprinkled in” [24]. True to this philosophy the Social Conference Manager was deliberately kept very simple. However, the tool is proper for further discussions because many possibilities of the OpenSocial standard have been exhausted. One possibility, for example, are the different views that can be created. The Social Conference Manager has the following views:

Profile view In the profile view a simple text is shown, which is created with opensocial templates (see Listing 4.3). Unfortunately, the possibilities are very limited in this view, because orkut does not allow running

⁶<http://www.aptana.org/>

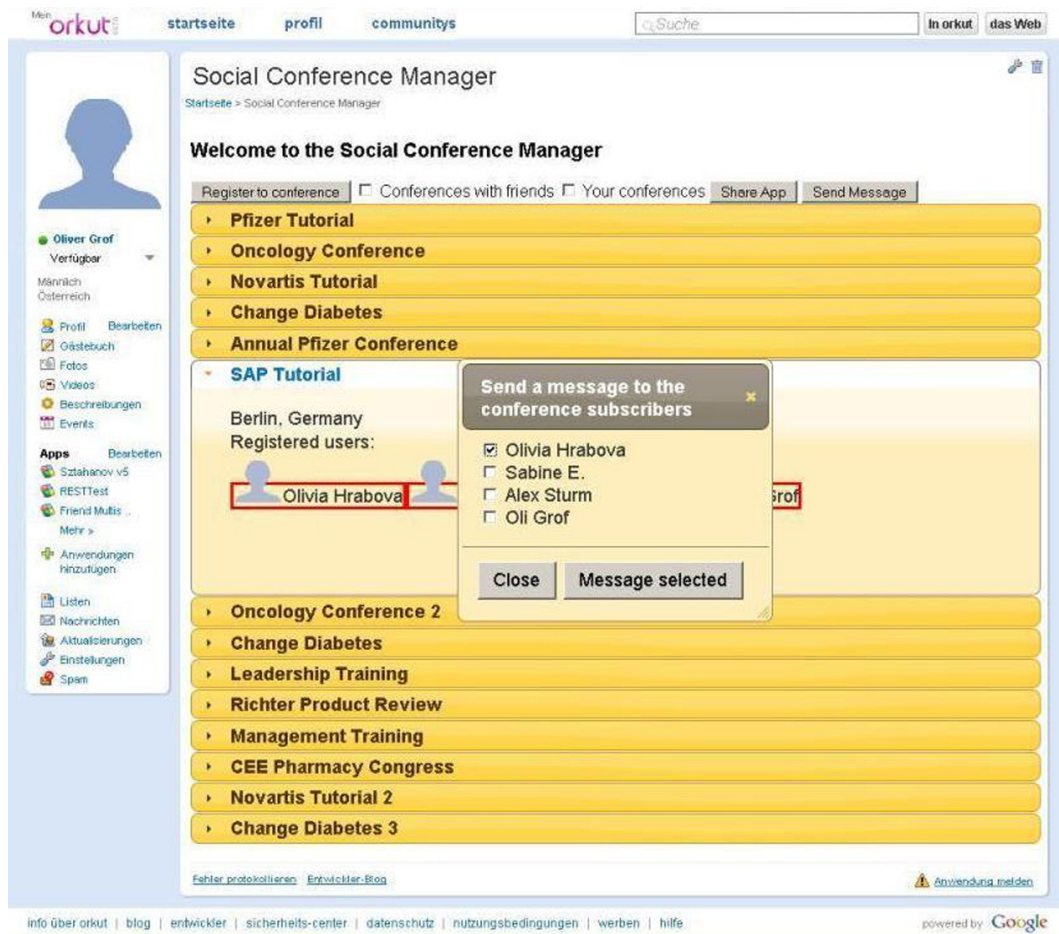


Figure 4.5: The Social Conference Manager developed for this thesis

JavaScript; otherwise all the conferences could be shown, to which the owner has been registered.

Canvas view After having installed the application, the features are available in the canvas view. If the viewer has not installed the application, the functionality is limited to reading conference information and seeing the registered users. In such a case, the user is notified about the limitations.

The following itemisation lists all the functionality of the implemented application, and compares the features with the corresponding use cases.

Register and deregister user If the user is the owner of the application, the tool adds a button, so that the user can register to or deregister from a conference. As it is written in the use case, she or he has to choose a conference first. After each (de)registration the user is asked whether she or he wants to post the current update on the activity stream.

Conferences of friends The viewer can select the conferences, where at least one friend is registered, via a checkbox.

Own conferences If the viewer clicks on the second checkbox, only those conferences are shown, where she or he has registered to.

Invite friends The viewer can recommend the application to his or her friends. After the button has clicked, the viewer can select the addressees of the invitation.

Send message This is the realisation of the use case 'Send Message to Members', actually. The addressees can be selected from the list of registered users; the subject of the message is set to "About: [Conference name]".

A big issue was, that the use cases were created based on the official OpenSocial 0.8 specification, which was not fully supported by the orkut sandbox. The following enumeration compares orkut's support with the descriptions of the OpenSocial Wiki⁷⁸:

1. The invitation to friends can be sent by using the `opensocial.requestShareApp` method, but it is not implemented in the orkut sandbox.

⁷[http://wiki.opensocial.org/index.php?title=Opensocial_\(v0.8\)](http://wiki.opensocial.org/index.php?title=Opensocial_(v0.8))

⁸[http://wiki.opensocial.org/index.php?title=Opensocial.Person_\(v0.8\)](http://wiki.opensocial.org/index.php?title=Opensocial.Person_(v0.8))

deleteCount	
<i>viewerId</i>	varchar(50)
<i>deleteId</i>	varchar(50)
<i>container</i>	varchar(20)

Figure 4.6: The table deleteCount

2. According to the documentation, the first argument of the method `opensocial.requestSendMessage` can be an ID, an array of IDs, or a group reference. However, in orkut it is not supported to send a message to multiple users. Therefore the Social Conference Manager has a serial message sending, more precisely multiple user selection is allowed, but the message has to be entered separately for each user. Due to this issue the “Send Message to Application Users” has not been implemented.
3. Creating links to the users’ profiles is not possible, because `opensocial.Person.Field.PROFILE_URL` is not implemented.

Lifecycle events (e.g., `event.removeapp`), which are also important and useful parts of the OpenSocial specification, are not supported in orkut, either. If a user, however, removes an application, his or her registrations must be deleted from the application server. This can be done with the following workaround. Normally, the conference data (including the ID’s of the registered users) is loaded and then the person’s data is requested from the container. In orkut the method `opensocial.DataRequest.newFetchPersonRequest`, which is used in this scenario, returns the user’s data only in case he or she has installed the application. If a particular user has uninstalled the application, then consequently no user information is returned. In such a case a delete-request is sent to the backend. Since this call is from a person who is not affected, a malicious user could launch an artificial query to delete the registrations of another user. Therefore, all delete requests are saved in the additional database table **deleteCount** (see Figure 4.6). The user data is only deleted irrevocably, when more than 10 requests from different users are in the table. If a user reinstalls the application and registers for a conference, before 10 requests have saved, then all the corresponding requests are deleted.

```

1 <Content
2     type="html"
3     view="profile"
4     preferred_height="100"
5     preferred_width="800"
6     refreshInterval="0">
7
8     <![CDATA[
9     <script
10         type="text/os-data"
11         xmlns:os="http://ns.opensocial.org/2008/markup">
12         <os:OwnerRequest key="owner" />
13         <os:ViewerRequest key="viewer" fields="name"/>
14     </script>
15     <script
16         type="text/os-template"
17         xmlns:os="http://ns.opensocial.org/2008/markup"
18         xmlns:osx="http://ns.opensocial.org/2009/extensions">
19
20         <span class="{viewer.gender}">
21             <b>Welcome {viewer.name.givenName}</b>
22         </span>
23         <span if="{viewer.id != owner.id}">
24             <a href="{owner.url}">
25                 {owner.name.givenName} {owner.name.familyName}
26             </a> has
27         </span>
28         <span if="{viewer.id == owner.id}">
29             You have
30         </span>
31         installed the Social Conference Manager: The application to manage
32         conference and socialize.
33     </script>
34     ]]>
35 </Content>

```

Listing 4.3: The profile view of the Social Conference Manager developed for this thesis

Chapter 5

Interview Analysis

In Chapter 4 OpenSocial has been introduced and the application called Social Conference Manager has been documented. The application has been deliberately kept simple, and merely serves as a basis for further discussion. The present chapter analyses the interviews, that have been made with selected users. First, the the course of the interviews is explained. Second, general requirements, that have been derived from the interviews are presented. Finally, opportunities for further development of the Social Conference Manager are listed.

The interviews were made with three selected users, and all of them represented different age groups and different occupational groups. Although the number of interviewees was low, this kind of diversity was very beneficial for the analysis, since it was very useful to see different viewpoints. Before the interviewees were questioned, they had received an information sheet to become familiar with the most important terms related to online communities (Section 2.4). The first part of the interviews were made with intent to find out the users' opinion on social software in enterprises and the WebOS (see Section 3.3) approach. Then the users were logged in to the orkut sandbox with previously created, individuals accounts. After a short introduction of the social networking portal, the users were asked to choose the Social Conference Application, so that they were able to test it. The testers clicked through the application and narrated their activities. Due to the fact, that the application was evaluated right in the second part of the interview, additional documentation of the test (e.g., with a video camera) was not worth the effort. As already mentioned above, the application has been deliberately kept simple anyway.

The interviews, which were made based on the guideline below, can be found

in Appendix A.

Usage of Social Software

- To what extent do the employees in your company communicate with each other? To what extent do they use social software to communicate with each other?
- How is the cooperative work coordinated? Are events and appointments, for example, planned together?
- Have you ever used social software? If yes, why and in which context? If no, why not?
- Are you going to use social software? If yes, why and in which context? If no, why not?

About the Application

- What are the positive/negative characteristics of the application in your opinion? Were you able to use the application easily or were there some usability issues?
- Would the use of this and similar applications in businesses make sense? Would it be possible to increase the effectiveness in the company in this way?
- What minimum requirements must be fulfilled that you use the application? What are your requests to an improved version of this software?
- Is it useful to design individual applications for specific problems, or is it sufficient to solve these problems with older or more abstract concepts (e.g., simple Groups, MS Outlook + MS Exchange Server, ...)?

5.1 Derived Requirements

To get to know different viewpoints of social software, three users of different occupation and different age have been asked. Although such a low number of interviewees cannot be considered as a representative sample of knowledge workers, some interesting trends can be found. For example, it became clear that the younger an interviewee is, the more open-minded is he or she about

social software in enterprises. However, Enterprise 2.0 concepts have not found the way to many companies, yet. Emails and mobile phones still work, events and appointments can be managed with Microsoft Outlook and Microsoft Exchange Server. Critics say it is too risky, too expensive to reform the business culture and adopt social software. Furthermore the efficiency-enhancing abilities are put into question. Even if users like new approaches, they have concerns about the successful introduction in the companies. The following subsections list the requirements that represent the “lessons learned” of the interviews.

Privacy Protection

Although people share a lot of information about themselves in open social networks, they are averse to do so in the enterprise. User 3 said the Social Conference Manager would not be very popular in her company, because it is not anonymous. Therefore a minimum requirement for an E2.0 platform is a clear data privacy protection. Users must know what other users see, who exactly sees a certain information and how the settings can be changed. More precisely, the configuration must be transparent for users. In addition it must be discussed whether or not it makes sense to allow anonymous interactions. Statement of User 3:

“It would of course be great to have a list of people who are interested in a particular subject and who may also have more experience. In the Social Conference Manager, for example, we could discuss about all the issues related to a particular conference. Parallel to this, we could go through the trainings on the intranet and discuss them together. However, the whole thing is not anonymous, thus everyone knew at once who is not familiar with a certain topic. In our business not showing such a weakness is crucial, therefore these concepts would not work.”

Management Support

Neither the reform of the business culture nor the use of new software packages can be successful without a committed management. Leaders of the company should support modern approaches, so that the knowledge workers consider these concepts as a part of their work. However, User 3 suggests rather a participation of employees than a dictatorial change. At the end of the interview

she said that a “*reformed business culture should be created together with the staff, and the rules should arise in the course of the negotiations*”.

Democratic Community

User 1 indicated that enforcing a strong hierarchy means relying on old concepts, “*which are not able to produce that kind of added value*” Enterprise 2.0 products are supposed to produce. Therefore a democratic community should be built, where every actor is able to contribute and the number of limitations is very low. Michael Holakovský also emphasises the term “community” when he criticises the restrictions of the the Austrian *Enterprise 2.0 Initiative*¹ in his blog entry².

Reasonableness

The community, the management and even the software developers are responsible for satisfied users, who are not overwhelmed by modern E2.0 concepts. Developers must care about usability issues right from the design phase and the management must offer appropriate tutorials for the potential users. The employees should be advised that the new tools do not mean extra work, but are part of the work. Today a number of knowledge workers would probably ask questions like User 3: “*By the way, should these applications all be used in our working hours?*”. This can be changed by providing appropriate tools. In some situations it might be very advantageous to integrate modern solutions into existing platforms. Last but not least, the community should be more open-minded about computer-supported social interactions in the enterprise.

Usability and Social Aspects

The requirements “privacy protection” and “reasonableness” highly rely on good usability. Both changing the settings of the data privacy protection and the use of modern tools must be straightforward for the users. Respective the requirements “management support” and “democratic community” it must be mentioned that not only technical issues but also social aspects have to be analysed. User 3, for example, would not dare to interact with colleagues on the social network and would prefer a phone call - or maybe video conference - instead. She said “everyone knew at once who is not familiar with a certain

¹<http://www.e20initiative.at/>

²<http://www.enterprise2punkt0.at/2010/05/kommerzieller-e2p0-blog.html>

topic”, if she would use the Social Conference Manager to ask question about a certain topic. Thus, taking the fear out of social software usage in enterprises is an important part of the change process, and one important step could be to describe the business culture right in the corporate agreement.

Before OpenSocial will be discussed in detail in chapter 6, the next section lists opportunities for further development of the Social Conference Manager and OpenSocial applications in general.

5.2 Opportunities for Further Development

Every software tool has opportunities for further development, and a few new requirements can be found out in the test phase. The Social Conference Manager has been tested by three users of very different occupation and age. The following subsections list the most important opportunities for further development based on both the user interviews and further considerations.

Add and Remove Conferences

In the current version of the application the conference data is not managed by any user, but by an administrator. In some cases, however, it could make sense to allow every user to perform these tasks. Traditionally, Web 2.0 products are characterized by increased user-generated-content and the harnessing of collective intelligence. Ideal Web 2.0 tools are open and democratic, so that every participator has unlimited possibilities of contribution. On the other hand, the interviewees have pointed out, that it is necessary to have a minimal hierarchy in all enterprises. When I asked User 1, who is a Sharepoint key user, about that hierarchy issue, his answer contained the following: *“Of course you have to be aware that the most companies need to be treated differently than the Web in general. Due to security issues and legal reasons some restrictions are unavoidable”*. Therefore the necessity of restrictions really depends on the context.

Anonymous Usage

For some employees it would be a big dilemma to use the Social Conference Manager because it is not anonymous. On the one hand Enterprise 2.0 is about a reformed business culture, where added value is derived from human social behaviour. It would be desirable that new software packages supplement social

interactions in real life, but do not replace them. On the other hand, it can be considered to let users create a nickname for some scenarios or post messages anonymously.

Platform Integration

An “issue is to suit social software to the work context and industrial business practices” [29, p.382]. There are many large corporations that have installed Microsoft Sharepoint as an Enterprise 2.0 solution. Due to the fact the employees have already absolved a number of trainings and tutorials, no chief information officer is likely to support another technology. However, it would be advantageous to combine some resources - reminiscent of some WebOS approaches - with the OpenSocial standard.

Localization

*“Internationalization and localization are terms that are frequently used (as synonyms) to mean that an application is set up to adapt to different regions and languages. Specifically, localization is the process of loading into the application different locale-specific text and elements. Sometimes the terms are mashed together into a numeronym called **i18n** and **L10n**.”* [33, p.267]

Gadgets can be localized with multiple message bundles, that are all referenced in the <Locale> tag. Developers may use the *osi18ntool* or the *hi5 message bundle translation service* for localization.

Adobe Flash Update

Many OpenSocial applications hosted by open social networks are Flash-based, therefore these applications have more beautiful and more dynamic user interface. It would be perhaps nice to have a world map, in which all the conference locations are marked. On the other hand, the information is much more important in an enterprise, than a perfect presentation. Of course the user interface must be straightforward, but we must not forget that the target group in enterprises differs from the target group in the Web. Nowadays the use of such applications is not considered as part of the work, anyway (see the interview with User 2 in Appendix A).

Mashups

AJAX applications can consume different services and combine different resources. The Social Conference Manager, for example, could have an integrated weather service and suggest the conference locations in the search form. Although it is not an easy job to find a simple service provider, it is possible to create such an updated version of the application. However, the combination of resources should not result in too complex tools. It is noticeable that an OpenSocial container already allows the creation of enterprise mashups. Users can add all the simple applications, which are required for their individual needs. If we combine too many features in one tool, it might become too complex, which is an issue in enterprise software (e.g., enterprise resource planning) these days.

The goal is to have a web-based platform (or web-based operating system) that allows the meaningful combination of resources and the elimination of unnecessary parts. A positive example has been told by User 1: “(...) managers can connect to a customer relationship management system via Sharepoint and can see only data, which is relevant for them”.

Chapter 6

Latitude for Enterprise 2.0

Having introduced the most important social software concepts and technologies in previous chapters, this chapter describes the potentialities of OpenSocial with respect to Enterprise 2.0. There are four issues that are clarified below. First, the relation of OpenSocial and Enterprise 2.0 is analysed, and the possibilities of OpenSocial in the enterprise are discussed. Second, potential alternative solutions are listed. Third, the relevance of social semantic web concepts is investigated. Finally, the fourth part points out some risks, and calls for a responsible usage of available technologies.

6.1 OpenSocial in the Enterprise

The applicability of OpenSocial in the Enterprise highly depends on the characteristics of the social network service, which acts as an OpenSocial container. An ideal container contains all the components and properties from the mnemonic FLATNESSES: *freeform, links, authorship, tags, network-oriented, extensions, search, social, emergence, signals* (see 3.1 for details). A detailed view of these characteristics helps to determine how this condition is fulfilled in social network platforms these days.

Freeform, authorship Given the fact that social network platforms support authoring by granting every user easy access, these platforms are freeform - indifferent to formal organizational identities - as well. Normally there are no barriers to authorship: users can leave comments almost everywhere or simply choose “Like” or “Dislike”.

Social, emergence The social network platforms, especially OpenSocial applications, facilitate social interactions online. The social behaviour of a

large group of people often results in emergent collaboration. In other words, a high number of motivated users leads not only to good quality of content but also to an enhanced collective intelligence.

Links, tags, search As described in Section 3.1, the ability of the search technology can be enhanced by links and tags. While linking of people and resources is essential for online social networks, tagging has not put through. The reason is that, unlike in blogs and wikis, the user generated content is much lower. However, meaningful tagging could be useful for shared multimedia.

Network-oriented OpenSocial applications are fully web-based and portable (reusable).

Extensions “Take tagging one step further by automating some of the work of categorization and pattern matching” [35, p.25]. One extension in terms of SLATES are friendship suggestions in social network platforms, but of course OpenSocial applications can also implement algorithms for pattern matching and use the user activity to create “smart” computers.

Signals In a social network *activities* are a suitable solution to notify users, if new content of interest appears. In OpenSocial activities can include direct emails to users, notifications, user updates and invitations (request to install an application) [33, p.130].

The detailed view of the properties above has shown, that social network services are very suitable Enterprise 2.0 tools, and OpenSocial, furthermore, enhances their ability. In certain circumstances it is meaningful to use the OpenSocial standard behind the firewall as well. Therefore many enterprise software vendors have already integrated social networking capabilities in their products and have created innovative OpenSocial-based solutions. This statement is consistent with the Google IO Developer Conference 2009¹:

“In the year and half since OpenSocial’s public launch, there are now over 700 million end users of OpenSocial applications across numerous social sites (containers) around the world. With OpenSocial’s proven global success in traditional social applications, the enterprise software community has now begun to realize its potential and build innovative solutions that cater to the enterprise.”

¹Google I.O 2009 - OpenSocial in the Enterprise <http://code.google.com/intl/en-EN/events/io/2009/sessions/OpenSocialEnterprise.html>

The IT concerns IBM, Oracle and SAP have all decided to integrate this new network programming standard in their existing platforms. The *IBM Mashup Center* supports the combination of content from multiple sources, including OpenAjax Widgets and OpenSocial Gadgets. It must be mentioned, however, that late 2009/early 2010 the OpenAjax Alliance (www.openajax.org) and the OpenSocial Foundation have been progressing towards integration of OpenAjax technologies into OpenSocial Gadgets. *Oracle Connect* allows people to share links, ideas, questions, code, etc. inside Oracle. On the one hand Connect is quite similar to an open social network, since it has a profile system, users can comment or like something and there is a kind of activity stream where everyone can read what is going on. On the other hand the dynamic filtering mechanism facilitates fast and efficient information retrieval, which is an important Enterprise 2.0 requirement. The code of Oracle Connect is JRuby on Rails, and Shindig acts as a container. SAP is a supporter of OpenSocial, because they are convinced that business processes can benefit from people's relationships. The Relationship Analysis Server allows to gather data from multiple sources (Exchange Server, HR system, CRM system, etc.) within the company. The *Social Network Analyser* application consumes the profile and the multi-relationship information - made available by the Relationship Analysis Server - through web services.

Many other companies besides these three huge concerns, have begun to built innovative solutions that cater to the enterprise. Salesforce (salesforce.com) as a leading vendor of customer relation management tools cares about modern enterprise collaboration as well. The *Salesforce Chatter* platform offers both state-of-the-art social networking capabilities and document collaboration features. In Chatter, users can receive updates from..

- colleagues, who have been previously chosen from the so-called corporate directory
- several (even external) applications
- shared documents, like spreadsheets, presentations etc.
- external social networks
- groups

The *eXo Platform* (www.exoplatform.com) is “an open source, open standard, Enterprise-scale portal, content management system and WebOS for Microsoft

Windows and Unix-like operating systems” [68] written in Java. The platform uses eXo Social to build a social network in the enterprise and hence to enhance collaboration. eXo Social supports application development using Google Gadgets and OpenSocial. Gadgets can be edited directly in the platform and added to a Dashboard. To reduce security risks, users are only able to use a Dashboard, if the access rights are set properly. eXo Social has, like the majority of the OpenSocial containers, an activity stream, where updates and new activities are posted.

The Australian software company Atlassian[®] uses the OpenSocial technology to increase the awareness of activity across software development teams [1]. The company has developed two major products, and both of them can host gadgets. *Confluence*[®] is an enterprise wiki collaboration software with gadget integration abilities, i.e. the gadget appears directly inline after the edited Confluence page has been saved [1]. The other major product of Atlassian is an issue tracking project management software named *JIRA*[®]. Similarly to eXo Social, JIRA contains a dashboard, where OpenSocial applications and other gadgets can be added. Besides being a social technology, Opensocial is also a technology stack for connecting a lot of enterprise applications. The JIRA dashboard, for example, can contain the Confluence QuickNav and the Confluence Network Updates, and developers can create an issue even inside Gmail with the JIRA Gadget.

Although OpenSocial has originally been designed for open social networks (also called consumer social networks) like orkut, MySpace and hi5, it holds great promise for social application development in enterprises. Besides the advantages introduced in the previous paragraphs, the following itemisation summarises common benefits of the OpenSocial technology.

Development Enterprises do not need to create an own API to integrate social abilities, activity streams or gadget support in their corporate platform. They can use the open, standardized, well documented specification instead.

Installation To see the gadgets in action, the applications must be added to the users’ profile. This is - in a well-designed container - a quite fast, easy process and not a real installation. Applications can be considered as web services, which implies that updates are accomplished rather on the backend than on the client side. Reinstallation of the workstation

does not mean reinstallation of all the software, because everything is running on the top of the browser. In view of these facts, OpenSocial applications are not affected by installation issues, such as described in Section 3.3.

Reusability Once a gadget has been written, it can be hosted on multiple OpenSocial containers. Developers can rely on the OpenSocial community and a company may use a number of already existing applications.

Single sign-on Users of modern Enterprise 2.0 products can adjust the platform to their individual needs, so they are able to create enterprise mashups. The main objective is to connect all the technologies, all the tools the knowledge workers need for collaborative work in the enterprise. An ideal web-based operating system authenticates the users with single sign-on, and the majority of the applications act as a service consumer.

6.2 Alternative Solutions

While the previous section has underlined the meaningfulness of OpenSocial usage in enterprises, the present chapter investigates alternative solutions. Since OpenSocial is a technological standard, the present section focuses on alternative technologies. First the demands on a potential technology are clarified, then portlets, widgets and the Facebook API is compared with OpenSocial.

Demands on Technology

Besides wikis and blogs, some knowledge workers already use open social networks to maintain their connections to partners in other companies and within the enterprise. The findings of Section 6.1 suggest that enterprise portals should be improved by the adoption of social networking capabilities and by allowing personalisation. The current trend are web-based operating systems, which are able to host a number of applications and can be customised by the user. If we simplify the concepts, such a WebOS-like platform can be built on two basic components:

Platform standard Applications can be developed based on a common standard. Enterprises could create own standard APIs, but they could also use open APIs like OpenSocial. Since Enterprise 2.0 concepts are not limited to the intranet, it would make sense to use standards that ease the emergent collaboration between businesses as well. However, a minimum requirement is a standard for a certain platform, so that the users can combine the relevant resources (=create mashups) themselves.

Container If a standard for a certain platform has been chosen or created, there is a need for an environment, which “understands” the products developed based on the standard. Therefore the standard specification has to contain the requirements of the container. If a container, for example, fulfils the requirements of the OpenSocial specification, then it is able to host OpenSocial applications.

Continuing the thoughts from above, Opensocial is an important piece of the puzzle, but it is not a stand-alone software solution. With other words, using additional technologies is essential for OpenSocial-based projects, therefore the standard cannot be compared with complete portal/WebOS solutions. However, it is not determined, which technologies should be used to implement

the backend. Even Apache Shindig², the reference implementation of the OpenSocial container, has a Java version and a PHP version. Version 1.1 of Apache Shindig has the following technical requirements³:

Java version “Servlet container supporting Web Application 2.3 or above and JDK 1.5 or above”.

PHP version “Apache Web Server with mod_rewrite enabled and PHP 5.2.x with the json, simplexml, mcrypt and curl extensions enabled”.

Besides the platforms mentioned in Section 6.1, Microsoft Sharepoint could be turned into an OpenSocial container as well. It is a topic of further research, whether or not the combination of Sharepoint Communities and OpenSocial is profitable.

Portlets and Widgets

Having clarified two basic components of modern web-based portals and possible backend solutions for OpenSocial, the present subsection deals with concrete alternatives. In actual fact, OpenSocial applications are special Google Gadgets, which support social interactions between the users of a social network. Potential alternatives for Google Gadgets are *Portlets* and several types of *Widgets*. The exact definition of portlets is included in the Java Portlet Specification 1.0 (JSR 168⁴) and the Java Portlet Specification 2.0 (JSR 286⁵).

“A portlet is an application that provides a specific piece of content (information or service) to be included as part of a portal page. It is managed by a portlet container, that processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to information systems.” [20, p.17]

A widget (**Window Gadget**) is a component of a graphical window system⁶. An exact definition of widgets does not exist, but the World Wide Web Consortium is working on a standard for web widgets⁷. However, there are many

²<http://shindig.apache.org/>

³<http://shindig.apache.org/download/index.html>

⁴Java Specification Requests 168 - Final Release: 27 Oct, 2003

⁵Java Specification Requests 286 - Final Release: 12 Jun, 2008

⁶<http://de.wikipedia.org/wiki/Widget>

⁷<http://www.w3.org/TR/widgets/>

different concepts these days - even Google Gadgets can be considered as widgets. A promising alternative of the Gadgets API is the Universal Widget API (UWA) for netvibes⁸. In contrast to portlets (and reminiscent of gadgets), the Universal Widget API has an OpenSocial extension, which eases the development of social applications. But which approach is the most appropriate for developing social enterprise applications? The standardisation process of portlets has already begun in 2002, therefore the standard is technically mature. On the other hand, the portlet specifications have not standardised social functionalities (e.g., managing people and their relationships). However, building corporate communities and allowing personalisation are basic requirements for Enterprise 2.0. What about UWA? Although UWA is an alternative for Google Gadgets, it is not an alternative to OpenSocial.

Although OpenSocial is a relatively new standard and version 1.0 is not supported by many containers these days, it is a very recommendable approach for Enterprise 2.0 solutions. Besides the technical benefits listed in Section 6.1, OpenSocial has following advantages:

- Google support - initially the API was developed by Google, now the corporation is represented in the OpenSocial Foundation.
- Large container repertory, huge community - a number of open social networks (e.g., MySpace, hi5, LinkedIn, Netlog, Ning, XING, iWiW) support the standard, therefore the developer community is very large.
- Support in enterprise platforms - IBM Mashup Center, Oracle Connect, eXo Platform, Salesforce, JIRA.

These advantages have led to a more straightforward, light-weight JavaScript API (namespace `osapi`) - as of version 1.0 the older Javascript API (namespace `opensocial`) is deprecated⁹.

6.3 Social Semantic Web

Latest findings and research (e.g., [19], [7]) make clear that modern web solutions include not only social facets but also semantic aspects. Sir Timothy

⁸http://dev.netvibes.com/doc/universal_widget_api

⁹<http://opensocial-resources.googlecode.com/svn/spec/1.0/Social-Gadget.xml>

John Berners-Lee, who is one of the most-cited authors in computer science, considers the following equation as a good description of the present evolution:

$$\text{Web 2.0} + \text{Semantic Web} = \text{Web 3.0}$$

Due to the fact that both Web 2.0 and Web 3.0 are buzzwords with no standard definitions, using the terms *Social Web* and *Social Semantic Web* might be more meaningful. The main goal is the enrichment of web content with explicit semantics so that the Web can be processed by machines [7, p.147]. An important concept, in this regard, is *Linked Data*, which principles were actually outlined by Tim Berners-Lee [19, p.60]:

1. Use Uniform Resource Identifiers (URIs) to identify things.
2. Use HTTP URIs so that these things can be referred to and looked up in the Web.
3. Provide useful RDF¹⁰ data about the thing when its URI is dereferenced.
4. Use links to reference to other, extern URIs, so that the user can discover other related things.

In recent years, a number of data providers have published geographic data or information about persons, scientific publications, books, online communities etc. as Linked Data, so that a global data space has arisen [19, p.61]. In such a data space it is possible to implement *semantic mashups* like generic data browsers, topic-specific portals and semantic search engines. [7, p.260]. As explained above, one principle of Linked Data is the use of the Resource Description Framework as a standard data model. If we want the semantic mashups to operate on data provided by our service, it is necessary to publish the data in RDF. There are many different notations to choose from; however, the RDF/XML representation is the most commonly used. Let's look at an example related to the application, that was implemented within the scope of the present thesis.

At the initialisation the Social Conference Manager receives the conference data as a simple XML. The XML document contains the root element, and the root element includes all the events (see Listing 6.1). It is human-readable and it can be parsed by the particular application it was designed for, but cannot be crawled by a machine agent. However, the data would become

¹⁰Resource Description Framework: the data model for the Semantic Web standardised by the World Wide Web Consortium (W3C)

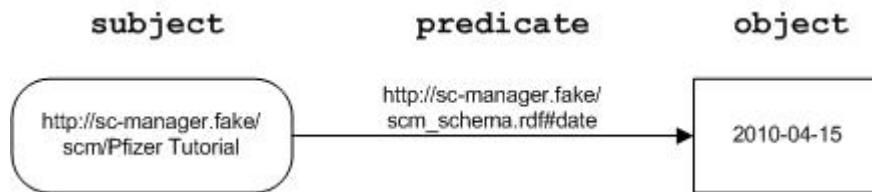


Figure 6.1: An RDF-triple as a directed graph

machine-readable by using RDF. Listing 6.2 shows a possible RDF/XML document, that identifies the entities of the original data. Besides the schema for the `rdf-syntax` in line 2, the document includes two namespaces: `scm` and `osoc` (lines 3 and 4). The namespace `scm` is a mini RDF schema designed especially for this example and the complete code can be found in Listing 6.3. The namespace `osoc` is the RDF schema of the term sets developed by the OpenSocial community. “OpenSocial/RDF can be used to enable and prepare requests to/from OpenSocial Containers, but also as a rich vocabulary for RDF-based social networking systems.”¹¹

Note Although the list of user registrations or any other information about the user might be a sensible data, it is included in the current example. In a productive environment, however, this would be serious issue.

The document in Listing 6.2 is indeed more complicated than the original document, but it can be part of the global data space in this form. The global data space consists of a set of linked triples, whereas a triple consists of a subject, a predicate and an object. Based on line 7 and line 9, for example, the directed graph in Figure 6.1 can be created.

Web resources in the Semantic Web become meaningful for machine agents, if we define vocabularies in RDF and identify (and link) resources with unique URIs. Let us suppose a class `event` exists in the data space, that has the subclasses `conference` and `tutorial`. The Oncology Conference, which is described from line 19 to line 30 in the exemplary RDF File, could be defined as a `conference` and could have a unique identifier (e.g., `http://sc-manager.fake/scm/Oncology Conference`). Regardless of the connection which would exists to the conference (e.g., a registration of a particular user), exactly the conference with that particular URI would be referenced. Similarly, people could be unambiguously identified, even if they had accounts for different social network platforms. One possibility is the creation of a FOAF¹² descrip-

¹¹<http://web-semantics.org/ns/>

¹²Friend-of-a-Friend <http://www.foaf-project.org/>

tion, which can be done computer-based¹³ as well. However, many other social semantic web approaches exists, for example Knowee¹⁴. A topic for further research is the combination of such approaches and other modern concepts (e.g., OpenSocial and OpenId¹⁵). However, some legal aspects and security issues must be clarified first. It also raises the question of whether it is desirable to uniquely identify users in the Web. In the interviews (see Appendix A), people were more likely skeptical. From a technical point of view, the enhanced structure and ability to response to complex requests are reasons to commend Semantic Web.

```
1 <data>
2 <event>
3     <name>Pfizer Tutorial</name>
4     <date>2010-04-15</date>
5     <place>Budapest, Hungary</place>
6     <users>
7         <user>06057222777039363515</user>
8         <user>17092804177187022092</user>
9     </users>
10 </event>
11 ...
12 </data>
```

Listing 6.1: XML representation of the conference data

¹³see FOAF-a-matic <http://www.ldodds.com/foaf/foaf-a-matic>

¹⁴<http://knowee.org/>

¹⁵<http://openid.net/>

```

1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:scm="http://sc-manager.fake/scm_schema.rdf#"
4     xmlns:osoc="http://web-semantic.org/ns/opensocial#">
5
6 <rdf:Description
7   rdf:about="http://sc-manager.fake/scm/Pfizer Tutorial">
8     <scm:name rdf:datatype="string">Pfizer Tutorial</scm:name>
9     <scm:date rdf:datatype="date">2010-04-15</scm:date>
10    <scm:place rdf:datatype="string">Budapest, Hungary</scm:place>
11    <scm:user>
12      <rdf:Bag>
13        <osoc:nickname>Oliver</osoc:nickname>
14        <osoc:nickname>Peter</osoc:nickname>
15      </rdf:Bag>
16    </scm:user>
17 </rdf:Description>
18
19 <rdf:Description
20   rdf:about="http://sc-manager.fake/scm/Oncology Conference">
21     <scm:name rdf:datatype="string">Oncology Conference</scm:name>
22     <scm:date rdf:datatype="date">2010-05-16</scm:date>
23     <scm:place rdf:datatype="string">Hamburg, Germany</scm:place>
24     <scm:user>
25       <rdf:Bag>
26         <osoc:nickname>Oliver</osoc:nickname>
27         <osoc:nickname>Peter</osoc:nickname>
28       </rdf:Bag>
29     </scm:user>
30 </rdf:Description>
31 </rdf:RDF>

```

Listing 6.2: RDF representation of the conference data


```

1 <?xml version="1.0" ?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
4
5 <rdfs:Class rdf:ID="Event"/>
6
7 <rdfs:Datatype rdf:about="&xsd:string"/>
8 <rdfs:Datatype rdf:about="&xsd:date"/>
9
10 <rdf:Property rdf:ID="name">
11     <rdfs:domain rdf:resource="#Event"/>
12     <rdfs:range rdf:resource="&xsd:string"/>
13 </rdf:Property>
14
15 <rdf:Property rdf:ID="place">
16     <rdfs:domain rdf:resource="#Event"/>
17     <rdfs:range rdf:resource="&xsd:string"/>
18 </rdf:Property>
19
20 <rdf:Property rdf:ID="date">
21     <rdfs:domain rdf:resource="#Event"/>
22     <rdfs:range rdf:resource="&xsd:date"/>
23 </rdf:Property>
24
25 <rdf:Property rdf:ID="user">
26     <rdfs:domain rdf:resource="#Event"/>
27     <rdfs:range
28         rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"/>
29 </rdf:Property>
30
31 </rdf:RDF>

```

Listing 6.3: Exemplary RDF schema file developed for this thesis

6.4 Assuming Responsibility

Independent of the new technology's type, minimising risks is always a challenge. However, modern network communication brings both opportunities and risks to the company. The present section discusses some issues in respect

of Enterprise 2.0 products, especially social networks. Before the discussion, reading the following thoughts on network communication is worthwhile:

“Companies should aim for control, but not too much, as a healthy variety and capitalizing on the economy of network stability are at least as important. A network is both a way to influence and to be influenced. Therefore, business relationships should be treated as first-class citizens; not just as a strategic tool to reach a given objective, but also as channel through which the company itself is formed. As such, listening, reflecting and reacting to others become central activities.” [61, p.50]

The goal of the studied concepts is, thus, to increase the effectiveness by creating a reformed, network-based business culture. Due to the fact that a high level of user participation is crucial, Enterprise 2.0 products need to be not only “freeformed”, linked etc., but also reliable. In terms of social aspects a concept is reliable, if it supports the work, so that privacy is not violated. However, social software poses some risks in this respect.

The interviewees (see Appendix A) have named the fear of **unwanted surveillance** in the interviews. Posting updates on the activity stream means that colleagues (and supervisors!) are informed, when exactly a certain employee was active and what exactly he or she has done. For legal reasons, trade unions are strictly against such solutions as well. Although these unions are certainly right, the basic problem is that many new approaches are considered as additional work, or sometimes even just an additional activity, which is not part of the work. While Enterprise 2.0 concepts aim inter alia for personalisation, employees prefer a clear separation of private life and work. Furthermore knowledge workers are likely to claim a **trustful data management**. Employees must be informed unambiguously about the access rights, the restrictions and the data protection. Realistically, it will certainly take years till users get accustomed to concepts such as cloud computing and web-based operating system.

Due to the lack of technology skills, a number of users is not familiar with the new opportunities. On the one hand it is desirable to have many users, who all participate in the emergent collaboration [35] by creating content themselves. On the other hand some evil groups have the possibility of **malicious manipulation**. Social bookmarking systems are certainly very useful, but are not

safe from malicious users - also known as spammers. Therefore BibSonomy¹⁶ has an integrated framework, which uses methods of machine learning to classify users [19, p.57]. The essay “Campaign 2.0: Unscrupulous Creation of Political Awareness?”, which can be found below, analyses the election in the USA and in Hungary, and points out some disadvantages of modern, Internet-based campaigns. As the concrete example shows, the Web is more affected by these dangers, as an enterprise software. However, social software is not necessarily limited to the intranet, and knowledge workers should keep a critical eye even within the company.

Software architects and developers call attention to some technology-based **security issues**. Since mashups, for example, consume a number of services, the service providers must be absolutely trustable. Furthermore the access rights of each tool or application must be unambiguous. OpenSocial containers ask users to set the preferences, when adding a new application to the profile, and allow them to change the preferences, after the application has been added. In addition, the present work has mentioned the oAuth protocol, which is able to secure the communication between different entities. However, there are many other security issues that cannot be solved with oAuth. In some cases issues force the OpenSocial community to change the standard, or the social network platforms to build in some restrictions. Message sending to multiple recipients is maybe disabled in orkut, because it might be used for sending spam.

Although specialists of Internet security have many proposals to minimise risks, assuming responsibility is unavoidable. Users should be aware of the environment, they are living and working in, and make use of the protection abilities of a certain technology.

Campaign 2.0: Unscrupulous Creation of Political Awareness?

Researchers and developers of advanced technologies must be always aware of the implications of new technology usage. There is already a series of publications which examine social aspects of computerization. In this regard, both Enterprise 2.0 and relevant topics are extremely critical because they focus on social interactions. The present thesis has already shown some risks and

¹⁶A web-based social bookmarking system for managing websites and publications.

calls for a responsible use. The present article shows what the creation of political awareness in the modern web could look like, and what the risks and possibilities are.

Barack Obama and the Web

Nowadays, the concepts of Web 2.0 (which are valuable Enterprise 2.0 approaches as well) are also used for political campaigns. An often-cited example is certainly the election of Barack Obama as President of the United States in 2008. His team managed not only a bottom-up donation movement, but also lined up the "Campaign 2.0". The former presidential candidate (or his team) has used virtual social network tools to be in touch with the supporters, in addition his YouTube channel¹⁷ and some podcasts¹⁸ personalised the campaign. As part of BarackObama.com the *Organizing for America Blog* started on February 9 2007.

Campaign 2.0 in the Hungarian Election

Motivated by the successful use in America, the political actors in Hungary became also aware of the tools listed in the previous paragraph. All parties that have a realistic chance of receiving five percent¹⁹ of the votes are present on the social network Facebook with relevant groups and profiles of top candidates. Having won over 14.000 fans within 30 days, Viktor Orban, the leader of the main opposition party Fidesz²⁰ seems to be particularly successful. In addition to the top candidate's page and various groups, the party has developed a simple but striking countdown application. The user can - reminiscent of other Facebook applications - also recommend it to friends and make the output appear in the activity stream. As is well-known, Enterprise 2.0 is not only about new software packages, but also about a reformed business culture. In this respect, the young group called LMP²¹ is quite interesting. As a kind of green party, they believe in a minimal hierarchy, have an open campaign account, and call their Web 2.0 offer LMP 2.0. Although an outsider cannot verify much of it, this philosophy has certainly potential for voters with Internet access, due to the desire for true transparency and freedom of expression. However,

¹⁷<http://www.youtube.com/user/BarackObamadotcom>

¹⁸<http://obama.podcast.com>

¹⁹The threshold in the Hungarian election is five percent nationwide.

²⁰Fiatalkor Demokraták Szövetsége (English: Alliance of Young Democrats)

<http://fidesz.hu/>

²¹Lehet Más a Politika (English: Politics can be different) <http://lehetmas.hu/>

the abuse of these new technologies can rapidly lead to disappointment, in extreme cases, democracy is threatened. Blogs can now be created in minutes and filled with almost any content. In certain circumstances, the comment feature - which is available not only in blogs, but also in the most news portals - can be very risky. Again and again it is revealed, that some parties have an army of volunteers with about 3-4 or even more virtual identities per person, and that these identities are used for one purpose only: malicious agitation in the comments. The motto of these groups is “A lie told often enough becomes truth”, if we can trust this piece of information. Especially the right-wing Jobbik²² and post-communist MSZP²³ are accused of such a crime, but the Internet allows anyone to manipulate collective intelligence freely and nearly anonymously.

Good or Bad?

It is now natural to ask whether the overall contribution of the Internet to an open, democratic life is positive or negative. On the one hand, certainly new doors have been opened; political parties can reach much larger number of voters much faster, and politicians can be monitored better now. The progress, therefore, seems to be very positive: while the time efficiency and structure of these media allow a more penetrative discussion of some issues, the cost efficiency and the control function can indeed contribute to the reduction of corruption. On the other hand, the Internet offers almost endless opportunities for manipulation, more precisely, advantages imply additional disadvantages. University graduates are encouraged to maintain a scientifically-critical view of the world. It would be desirable that all eligible voters verify the information thoroughly before making a choice. Of course it is also possible to lie in print media, television and radio, but due to the lower number of actors, information in these media are easier to verify. The flood of information and endless possibilities represent the greatest challenges e-democracy will have to face. In 2010 the Internet will not be a decisive factor in elections in Hungary because just a few users are reached through this channel. What about 2014? Will the voters have a strong relation to reality or will the (malicious) Web be strong enough to manipulate them?

²²Jobbik Magyarorszagert Mozgalom (English: Movement for a Better Hungary)
<http://jobbik.hu/>

²³Magyar Szocialista Part (English: Hungarian Socialist Party) <http://mszp.hu/>
As of 2002 governing Party in Hungary

Chapter 7

Summary

In the present thesis, attempts have been made to summarise and assess the current research trends of social software in enterprises. The scaffold of the work is based on two major pillars: the theoretical discussion of Enterprise 2.0 and the technical study of OpenSocial. As a result of the exhaustive literature research and in-depth analysis the work has both a strong fundament and a solid structure. The following paragraphs summarize the main body (one paragraph per chapter), list the the limitations of the work and suggestions of further research (one paragraph each) and explain practical implications.

Due to the fact that the chosen research area cannot be examined independently of any other field, the related terms are discussed first. In this regard Computer Supported Cooperative Work (CSCW), business process management, knowledge management and Web 2.0 are relevant concepts, because all of them has influenced Enterprise 2.0. It was found that CSCW is the theoretical fundament of groupware, and furthermore two accepted classifications of groupware were introduced. A range of research areas and approaches rely on workflow management, which is major topic in CSCW. In Section 2.1 enterprise resource planning (ERP) and supply chain management (SCM) is mentioned, and Section 2.2 addresses business process management(BPM). The concepts of artifacts, however, connect CSCW and knowledge management. For knowledge management organisational memory and personal networks are important terms. Another large related research area is “Online Communities”, which contains Web 2.0, social software and social network services. In addition, Section 2.5 explains the term “enterprise mashup” and the concept’s connection to Service-Oriented Architecture (SOA) and Software as a Service (SaaS).

Second, Enterprise 2.0 concepts, trends and technologies are introduced. The

definition of Enterprise 2.0, the conceptions SLATES and FLATNESSES and current trends are explained based on the literature research, that has been done within the scope of the present work. The analysis of selected case studies show that using collaborative authoring tools (wikis) and blogs, and supporting social networking might be meaningful in enterprises. Furthermore, modern software solutions (especially Microsoft Sharepoint 2007/2010) and usability issues are discussed and the most important underlying technologies are introduced. Section 3.5 examines how CSCW and Web 2.0 have influenced Enterprise 2.0. The objective is to clarify ambiguities by explaining the difference of groupware and social software, and by listing the definitions of network effect, communities of practice and communities of interest.

The Chapter *OpenSocial in Action* begins with a crash course of this new standard and shows two basic examples. Since OpenSocial applications must be hosted on an appropriate container, four social network platforms are compared in a separate section. Section 4.3 documents the most important facts in respect of the Social Conference Manager - the application, which was implemented within the scope of the present thesis. The documentation is divided into two sections: *Use Cases* and *Implementation Details*. The application has been deliberately kept simple, so that it could serve as a basis for further discussions.

Within the scope of the present thesis three users were interviewed with intent to find out their opinion on social software in enterprises. Another objective was to evaluate the prototype of the Social Conference Manager. The structure of an interview session was as follows: (i) interviewees received an information sheet that contains the most relevant information, (ii) the first part of the interview was made with general question about the usage of social software, (iii) the application was tested by the users, who narrated the their activities, (iv) the second part of the interview was made with question about the application. Chapter five contains the analysis of these interviews. As a result of the analysis four general requirements for Enterprise 2.0 could be specified: a clear data privacy protection, management support, democratic community and reasonableness of the tools. Furthermore, the opportunities for further development of the Social Conference Manager were discussed.

Chapter six connects the two mentioned pillars and thus completes the work. First, it analyses the relation of OpenSocial and Enterprise 2.0, and the possibilities of OpenSocial in the enterprise. It was found that social network

services are very suitable Enterprise 2.0 tools, and OpenSocial, furthermore, enhances their ability. A number of IT companies support OpenSocial in their enterprise portal solution. Section 6.1 analyses both the major concerns, like IBM, Oracle and SAP, and other innovative companies, like eXo, Salesforce, Atlassian. The current trend are web-based operating systems, where all the applications are running on the top of the browser. Second, chapter six discusses potential alternative solutions: portlets and widgets. Due to the fact that OpenSocial standardises social functionalities (e.g. managing people and their relationships), it is the most recommended concept among the possible options. Third, the relevance of social semantic web concepts is examined. Section 6.3 includes not only a theoretical analysis but also investigates, which semantic aspects could be integrated into the Social Conference Manager. Based on the self-designed, exemplary RDF schema and the XML representation of the conference data is turned into an RDF data. The final part of chapter six points out some risks, and calls for a responsible usage of available technologies.

As always, the investigation had a number of limitations. The biggest issues were the current state of the OpenSocial standard (as of March 2010 v0.8 was the most supported version) and the restrictions of the social network platforms, that have been used as OpenSocial containers. The administrators of orkut, hi5, iWiW, etc. have considered some security risks, probably. Furthermore, businesses still highly rely on older communication channels, such as mobile phones and email; therefore employees do not consider Enterprise 2.0 approaches as part of their work.

Further work should be conducted to analyse the latest version of the OpenSocial standard, without having restrictions in the hosting container. Another line of research worth pursuing further is to study how OpenSocial can be successfully integrated into current platforms and how it can be combined with other technologies. It should be also determined whether the widespread enterprise platform, Microsoft Sharepoint can be redesigned as a web-based operating system. It should be found out whether or not there is a possibility of making the important concept Sharepoint Communities and OpenSocial work together. In addition the snapshot approach explained in Section 2.3 should be analysed further, since it might have big potential in research related to E2.0, social networks and OpenSocial.

Obviously, communication, social interaction and the close collaboration of

employees are crucial for the success of enterprises. Businesses could benefit from having social network services installed, in particular if some reusable applications could be combined in individual enterprise mashups. OpenSocial has turned out to be a good Enterprise 2.0 approach, since it contains all the components and properties from the mnemonic FLATNESSES: freeform, links, authorship, tags, network-oriented, extensions, search, social, emergence, signals. Besides the social network concepts of the leading IT corporations, the eXo Platform and the products of Atlassian are mentionable enterprise software packages that integrate OpenSocial. Due to convergency of Social Web and Semantic Web, both OpenSocial and Enterprise 2.0 in general will have to face the challenges of the Social Semantic Web. The present thesis contains some basic thoughts on this area as well (see Section 6.3).

However, Enterprise 2.0 solutions - reminiscent of the majority of new technologies - cannot be applied safely and successfully without a responsible usage.

Appendix A

User Interviews

Within the scope of the present thesis three people have been questioned. All interviews can be read below. Although the interviews have been made anonymous, the age and the occupation of the users have been documented.

The Interview with User 1

- Age: 29
- Occupation: employee - portfolio management, customer relationship management, Sharepoint key user

Oliver Grof (O): To what extent do the employees in your company communicate with each other? To what extent do they use social software to communicate with each other?

User 1 U1): Like in the majority of the concerns, communication is crucial for business success. The services and features of *Microsoft Sharepoint* are available at our company, but as far as I know the Web 2.0 and social software features are not used frequently yet. Although wikis, for example, can be created very easily and fast, I do not know any of my colleagues, who have used this possibility. However, Microsoft Enterprise Content Management (ECM) solutions are supported very much. Another social software used in our company is the business social network *XING*.

O: How is the cooperative work coordinated? Are events and appointments, for example, planned together?

- U1:** We are using the Microsoft Exchange Server together with Microsoft Outlook's Calendar for appointment coordination. The accounts of the employees are set up and maintained by the IT department. Furthermore some features of Sharepoint (e.g. voting) are quite useful and popular.
- O:** Have you ever used social software? If yes, why and in which context? If no, why not?
- U1:** Since XING seems to be very popular, our department decided to send an event's invitations via this platform to the partners of the company. For that reason, I have signed up for this social network platform. Unfortunately, many of our partners did not have an account, which forced us to send the invitations through another channel. I asked XING for more information about business cooperation, but I received their answer one and a half month later, which was too late for our project.
- O:** Are you going to use social software? If yes, why and in which context? If no, why not?
- U1:** In my opinion the major companies are only at the beginning of the Enterprise 2.0 way. I myself have also signed up for XING for one specific project and - as already mentioned- many Sharepoint features are not really widespread. Obviously, this is going to change in future, therefore I will probably use social software, too.
- O:** What are the positive/negative characteristics of the application in your opinion? Were you able to use the application easily or were there some usability issues?
- U1:** I was able to use the application easily; the user interface is clear and pragmatic. I have one question: Who is allowed to add and remove conferences? Are there only a few administrators or can any user perform these tasks?
- O:** Thank you for asking this question. In the current version only an administrator can add and remove conferences. Due to the fact that real Enterprise 2.0 software should be - reminiscent of Web 2.0 tools - open and democratic, I have considered allowing any user to perform these tasks. Do you think it would make sense to design the applications for enterprises that are open and that are democratic? Or do you agree with those who require (at least) a minimal hierarchy?

U1: It depends on the purpose of the product. Using an application in an open social network for private purposes differs from using it in the enterprise. It is, for example, not necessarily meaningful to let all the users add any event, even if these events have nothing to do with the work. On the other hand, the present application might be limited to pharmacological conferences, which would imply that the open, democratic approach would make sense. In my understanding, there is no added value of Web 2.0/Enterprise 2.0 without user generated content and social interaction. All those actors, who believe and enforce a strong hierarchy, rely on old concepts, which are not able to produce that kind of added value. Of course you have to be aware that the most companies need to be treated differently than the Web in general. Due to security issues and legal reasons some restrictions are unavoidable.

O: Would the use of this and similar applications in businesses make sense? Would it be possible to increase the effectiveness in the company in this way?

U1: In the current state I do not believe that such applications will be useful in business context, although the effectiveness can be enhanced.

O: Due to a several reasons, user participation is quite low, isn't it? Nowadays the use of social software is just more work without any real advantages.

U1: Well, we use many of Sharepoint's features quite frequently (creating websites with ECM, votings, ...), but using real Web 2.0 features are not widespread. And a number of colleagues have signed up for XING to manage their contacts.

O: What minimum requirements must be fulfilled that you use the application? What are your requests to an improved version of this software?

U1: Sharing the application with colleagues would be very useful.

O: The OpenSocial specification contains this method, actually. Unfortunately, the present container (orkut) has not implemented this feature actually.

U1: I see. A very important requirement for a corporate platform is single sign-on. In the current corporate platform the users can use the intranet right after the Windows logon, which is quite good. The problem in

enterprises is the need of restrictions, but managing access rights is quite easy in Sharepoint.

O: Is it useful to design individual applications for specific problems, or is it sufficient to solve these problems with older or more abstract concepts (e.g. simple Groups, MS Outlook + MS Exchange Server,...)?

U1: It is definitely a topic to design individual applications. For example managers can connect to a customer relationship management system via Sharepoint and can see only data, which is relevant for them. And there are many more applications that are available directly in Sharepoint.

The Interview with User 2

- Age: 35
- Occupation: software engineer

Oliver Grof (O): To what extent do the employees in your company communicate with each other? To what extent do they use social software to communicate with each other?

User 2 (U2): In principle, we only communicate via email, and accordingly we have company owned mobile phones, which we can use for urgent calls. We do not use social software.

O: Maybe the company, where you are employed, is too small, and it does not really make sense to use social software. How is the cooperative work coordinated? Are events and appointments, for example, planned together?

U2: Yes, we use the calendar in Microsoft Outlook to coordinate the collaborative work. Therefore every employee is asked to share his or her calendar; furthermore setting up an out of office reply is obligatory. So we use this tool very intensively.

O: Have you ever used social software? If yes, why and in which context? If no, why not?

U2: I have used social software for private purposes: to keep in touch with friends, for example. Currently, I use the social network platform Facebook and the VoIP (voice over IP) application Skype, which is very

advantageous to make international phone calls. However, I do not chat on Skype frequently.

O: Are you going to use social software in future? If yes, why and in which context? If no, why not?

U2: In private context I am going to use social software surely. Of course you want to have some peace and quiet sometimes, but on many other days you want to know something about your friends' activities. I think it is quite useful to have a platform for social interactions. On the other hand, I cannot say, whether or not I am going to use social software in business context.

O: You have mentioned Facebook. Do use any other social network platforms (e.g. XING)?

U2: I have several other accounts, but I do not update my profiles there anymore. In my opinion, XING is for professional purposes. I have an account there as well, but I do not use it frequently.

O: Let's talk about the Social Conference Manager. What are the positive/negative characteristics of the application in your opinion? Were you able to use the application easily or were there some usability issues?

U2: I was able to use the application quite easily, and there were no real issues. The user interface is straightforward, therefore users should be not be overwhelmed by the application. I have only one question. Can I add new conferences?

O: This feature is not implemented in this version; only administrators can create and delete conferences. Do think that everyone should be able to perform these tasks in an enterprise?

U2: Well, not everyone. However, there should be a group of people, who are allowed to perform these tasks, not only a few administrators. In my opinion even the most democratic enterprise needs a minimal hierarchy.

O: Would the use of this and similar applications in businesses make sense? Would it be possible to increase the effectiveness in the company in this way?

U2: If an enterprise has a social network platform in the intranet, these applications make definitely sense. The employees can search, install or remove all the applications they need; therefore effectiveness can be increased. I really believe that added value can derive from human social behaviour. For me as a developer it would be interesting to create small, but useful applications with a simple, standard API myself. So I would not only be an application consumer, but also an application provider.

O: What minimum requirements must be fulfilled that you use the application? What are your requests to an improved version of this software?

U2: First, the application should be running stable. Second, the user interface should be intuitive and users should not face unexpected situations. Third, a single sign-on platform would be great.

The Interview with User 3

- Age: 50
- Occupation: pharmaceutical officer, sales representative

Oliver Grof (O): To what extent do the employees in your company communicate with each other? To what extent do they use social software to communicate with each other?

User 3 (U3): We communicate via Email and via mobile phones, but we do not use social software. We use the Internet to research for our work. In addition our company has an intranet, where the employees can log in and use the offered services.

O: How is the cooperative work coordinated? Are events and appointments, for example, planned together?

U3: Rather the office staff, but I am a sales representative. There are situations that we have to synchronise our appointments and check each others activities. For international business management we have a number of teleconferences and many colleagues have also a webcam installed, so they are able to build a connection with the computer instead of the mobile phone. The intranet, on the other hand, contains some standard forms, a calendar and even online trainings. Each employee is able to add an event to the calendar or to complete an online training.

O: Have you ever used social software? Are you going to use such a software in future? If yes, why and in which context? If no, why not?

U3: No, I am not using social software these days and I am not planning to use such a software soon. But I have heard of plans of a platform for doctors. First the doctors would have to sign up with their email address, then they would be able to maintain their contacts and look for relevant informations. This solution would be linked to the company.

O: I see. Within the scope of my thesis I have read some publication about the so-called web-based operating system (abbreviated WebOS). In this approach, it is important that applications are no longer installed locally on your desktop, but on a web-based platform. In other words, the applications are running on the top of the browser. To simulate such a scenario, I have developed the Social Conference Manager application for the open social network service orkut. What are the positive/negative characteristics of the application in your opinion? Were you able to use the application easily or were there some usability issues?

U3: In my opinion, you become accustomed to the majority of such new tools, if you are working with them frequently. The present application is quite straightforward; I like the colours and the alignment of the different parts. Some portals are so confusing that I do not know my way around there anymore! There are, for example, large differences among the free e-mail providers. Some of them have good usability but some of them are really bad. The Social Conference Manager is nice, colours are not too lurid and the application has a clear user interface.

O: Would the use of this and similar applications in businesses make sense?

U3: Although the concept is good, I am afraid I would not use such applications in real life. It would of course be great to have a list of people who are interested in a particular subject and who may also have more experience. In the Social Conference Manager, for example, we could discuss about all the issues related to a particular conference. Parallel to this, we could go through the trainings on the intranet and discuss them together. However, the whole thing is not anonymous, thus everyone knew at once who is not familiar with a certain topic. In our business not showing such a weakness is crucial, therefore these concepts would not work.

- O:** Do you mean, these applications (and social network platforms in general) do not make sense because they are not anonymous? Can you complete the online trainings anonymously?
- U3:** No, the online trainings are not anonymous, either. The point is, that our community would be too small, therefore no one would admit weakness. By the way, should these applications all used in our working hours?
- O:** Yes, a well-designed system should support and optimise your work, actually. Do you think it is possible to increase the effectiveness in the company in this way or it sufficient to solve these problems with older concepts?
- U3:** The problem is that chief executive officers and bosses often expect the employees to be familiar with such technologies very quickly. Sales representatives, on the other hand, are using the computer quite seldom and have not time to become acquainted with these tools. Perhaps this approach is useful for other areas, but the effectiveness of our work cannot be increased by these tools.
- O:** What minimum requirement must be fulfilled that you use the application?
- U3:** Well, your supervisor and your colleagues all know exactly what you have done all the day, right? Therefore it is an extremely important requirement that I can decide whether or not to publish my activity on the activity stream. Orkut's solution, for example, is acceptable.
- O:** You remind me of user 1. He was said the trade union claim restrictions due to legal reasons. In their understanding the employees are kept under surveillance, which is illegal.
- U3:** Exactly, this is also an issue at our company, and they would be strictly against it for the same reasons. The problem is that every activity is marked with an exact date and time. This information can all be used against you.
- O:** A number of publications recommend not only new software packages but also a reformed business culture. Many Enterprise 2.0 projects fail because organizational leaders are not engaged in supporting these reforms.

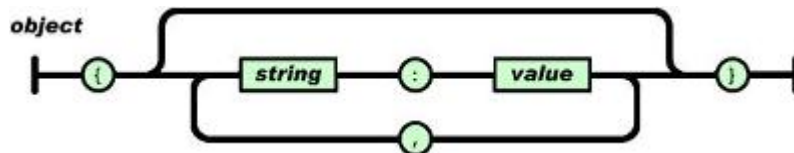
U3: Many employees are tired of all the suggestions of the bosses. A reformed business culture should be created together with the staff, and the "rules" should arise in the course of the negotiations.

Appendix B

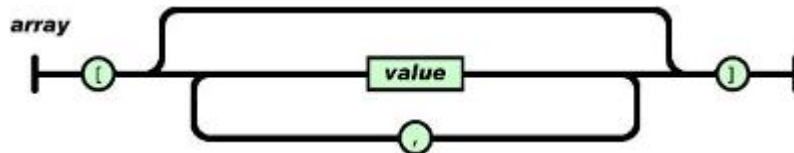
JSON Structures

Below there is a description of object, array, value, string and number based on the structures on which JSON is built. The content of this chapter has been extracted from [2], last visited on 4th February 2010.

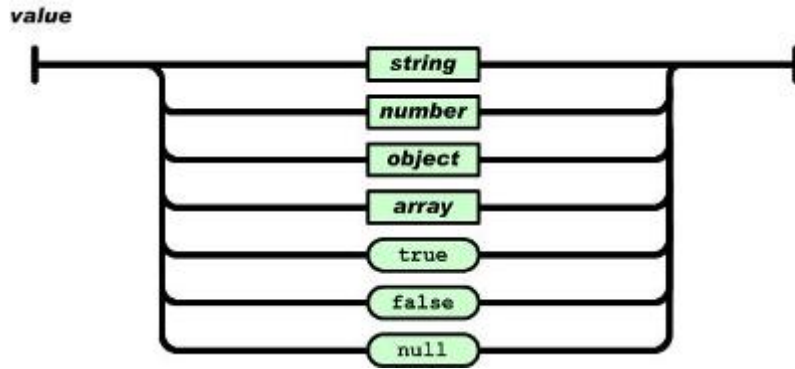
”An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).



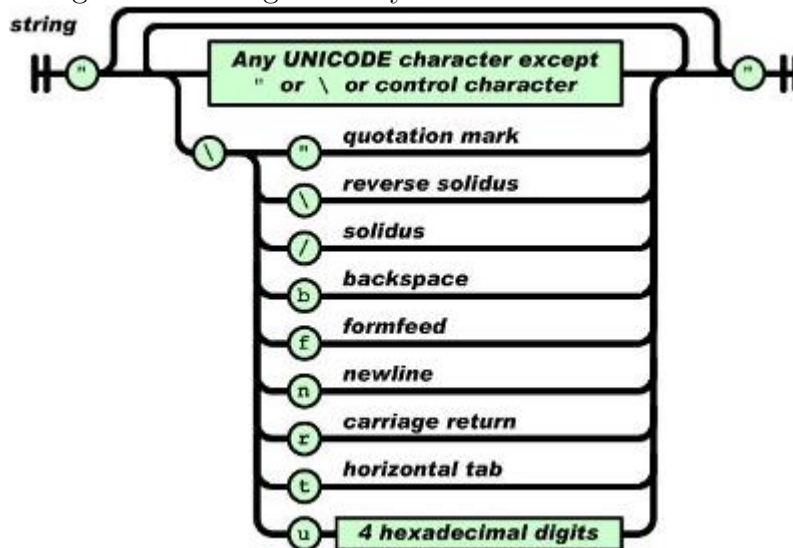
An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).



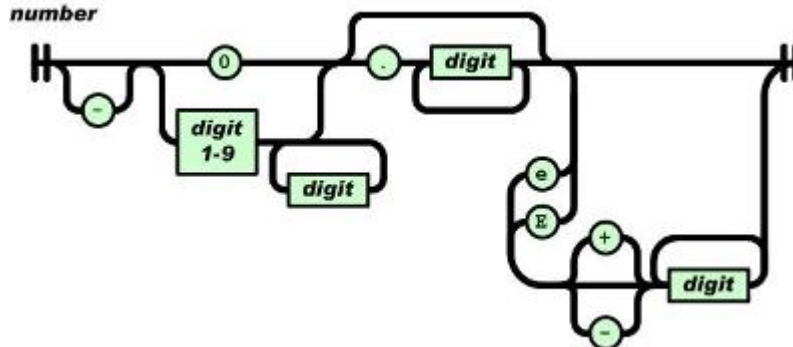
A *value* can be a *string* in double quotes, or a *number*, or *true* or *false* or *null*, or an *object* or an *array*. These structures can be nested.



A *string* is a collection of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.



A *number* is very much like a C or Java number, except that the octal and hexadecimal formats are not used.



Whitespace can be inserted between any pair of tokens. Excepting a few encoding details, that completely describes the language.”

Bibliography

- [1] *Atlassian+opensocial*. <http://www.atlassian.com/opensocial/>, Retrieved at: 25.02.2010.
- [2] *Introducing json*. <http://www.json.org/>, Retrieved at: 28.08.2009.
- [3] *Ergonomic requirements for office work with visual display terminals (vdts) - part 11 : Guidance on usability*, 1998.
- [4] *developerworks interviews: Tim berners-lee (transcript)*, July 2006.
- [5] M. ALUTOIN, S. LEHTONEN, J. HOEBEKE, G. HOLDERBEKE, I. MORMAN, L. SANCHEZ, J. LANZA, W. LOUATI, AND D. ZEGHLACHE, *Towards self-organising personal networks*, in DIN '05: Proceedings of the 1st ACM workshop on Dynamic interconnection of networks, New York, NY, USA, 2005, ACM, pp. 12–16.
- [6] J. K. BEATON, B. A. MYERS, J. STYLOS, S. Y. S. JEONG, AND Y. C. XIE, *Usability evaluation for enterprise soa apis*, in SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments, New York, NY, USA, 2008, ACM, pp. 29–34.
- [7] A. BLUMAUER AND T. PELLEGRINI, eds., *Social Semantic Web: Web 2.0 - Was nun?*, X.media.press, Springer, 2009.
- [8] J. CARDOSO, R. P. BOSTROM, AND A. SHETH, *Workflow management systems and erp systems: Differences, commonalities, and applications*, *Inf. Technol. and Management*, 5 (2004), pp. 319–338.
- [9] A. CHRITOPHER, *Tracing the evolution of social software*, *Live With Alacrity Blog*, (2004). http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html, Retrieved at: 29.07.2009.

- [10] T. COATES, *An addendum to a definition of social software*, January 2005. http://www.plasticbag.org/archives/2005/01/an_addendum_to_a_definition_of_social_software/, Retrieved at: 31.07.2009.
- [11] A. DI IORIO, D. ROSSI, F. VITALI, AND S. ZACCHIROLI, *Where are your manners?: Sharing best community practices in the web 2.0*, in SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing, New York, NY, USA, 2009, ACM, pp. 681–687.
- [12] R. DUNCAN AND A. WEISS, *Organizational learning: Implications for organizational design*, in Staw, B. (ed.). *Research in Organizational Behavior* (v.1), JAI Press, 1979, pp. 75–123.
- [13] C. A. ELLIS, S. J. GIBBS, AND G. REIN, *Groupware: some issues and experiences*, *Commun. ACM*, 34 (1991), pp. 39–58.
- [14] R. T. FIELDING, *Architectural Styles and the Design of Network-based Software Architectures*, PhD thesis, University of California, Irvine, Irvine, California, 2000.
- [15] G. FISCHER, *Communities of interest: Learning through the interaction of multiple knowledge systems*, in 24th Annual Information Systems Research Seminar In Scandinavia (IRIS'24), Ulvik, 2001, pp. 1–14.
- [16] L. FRANZ, *Wissensmanagement - Grundlagen, Methoden und technische Unterstützung*, Hanser, 2 ed., 2008.
- [17] V. FRIEDMANN, *Praxisbuch Web 2.0*, Galileo Press, 2. ed., 2009.
- [18] B. GRABOT, A. MAYÉRE, AND I. BAZET, *ERP systems and organisational change : a socio-technical insight*, Springer-Verlag London Limited, 2008.
- [19] U. HENGARTNER AND A. MEIER, eds., *Web 3.0 & Semantic Web*, no. 271 in HMD – Praxis der Wirtschaftsinformatik, dpunkt, Heidelberg, 2010.
- [20] S. HEPPER, *JavaTM portlet specification - version 2.0*, January 2008.
- [21] D. HINCHCLIFFE, *The state of enterprise 2.0*, October 2007. <http://blogs.zdnet.com/Hinchcliffe/?p=143>, Retrieved at: [Last visited:20090922].

- [22] —, *14 reasons why enterprise 2.0 projects fail*, August 2009. <http://blogs.zdnet.com/Hinchcliffe/?p=718>, Retrieved at: 30.09.2009.
- [23] —, *Sharepoint and enterprise 2.0: The good, the bad, and the ugly*, May 2009. <http://blogs.zdnet.com/Hinchcliffe/?p=280>, Retrieved at: 27.01.2010.
- [24] J. N. HOOVER, *Most business tech pros wary about web 2.0 tools in business*, February 2007. <http://www.informationweek.com/news/global-cio/showArticle.jhtml?articleID=197008457>, Retrieved at: 30.07.2009.
- [25] V. HOYER AND M. FISCHER, *Market overview of enterprise mashup tools*, in ICSSOC '08: Proceedings of the 6th International Conference on Service-Oriented Computing, Berlin, Heidelberg, 2008, Springer-Verlag, pp. 708–721.
- [26] D. JOAN, R. DAVID, G. WERNER, D. CASEY, B. BETH, AND M. MICHAEL, *Motivations for social networking at work*, CSCW'08, (2008).
- [27] R. K. L. KO, *A computer scientist's introductory guide to business process management (bpm)*, Crossroads, 15 (2009), pp. 11–18.
- [28] M. KOCH, *Cscw and enterprise 2.0 - towards an integrated perspective*, in Proc. Conf. Bled eConference eCollaboration, 2008, pp. 416–427.
- [29] T. KOSKINEN, *Social software for industrial interaction*, in OZCHI '06: Proceedings of the 18th Australia conference on Computer-Human Interaction, New York, NY, USA, 2006, ACM, pp. 381–384.
- [30] R. M. LERNER, *At the forge: Opensocial and google gadgets*, Linux J., 2008 (2008), p. 8. Retrieved at: 04.05.2009.
- [31] E. L. LESSER AND J. STORCK, *Communities of practice and organizational performance*, IBM Syst. J., 40 (2001), pp. 831–841.
- [32] J. LIEBOWITZ, *Knowledge management handbook*, CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431, 1999.
- [33] G. LYNNE, *OpenSocialTM Network Programming*, Wiley Publishing, Inc., 2009.

- [34] O. MARJANOVIC, H. SKAF-MOLLI, P. MOLLI, AND C. GODART, *Collaborative practice-oriented business processes creating a new case for business process management and cscw synergy*, in COLCOM '07: Proceedings of the 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing, Washington, DC, USA, 2007, IEEE Computer Society, pp. 448–455.
- [35] A. MCAFEE, *Enterprise 2.0: The dawn of emergent collaboration*, MIT Sloan Management Review, 47 (2006), pp. 20–28.
- [36] —, *Enterprise 2.0, version 2.0*, May 2006. http://andrewmcafee.org/2006/05/enterprise_20_version_20/, Retrieved at: 03.08.2009.
- [37] —, *Its not not about the technology*, July 2007. http://andrewmcafee.org/2007/07/its_not_not_about_the_technology/, Retrieved at: 03.08.2009.
- [38] K. MICHAEL AND R. ALEXANDER, *Enterprise 2.0: Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen*, Oldenbourg, 2007.
- [39] MICROSOFT, *People working together drive business results(white paper)*, 2009.
- [40] —, *The business value of sharepoint 2010 composites(white paper)*, 2010.
- [41] —, *Microsoft sharepoint 2010 - evaluation guide for technical and business decision makers*, 2010.
- [42] A. MISLOVE, M. MARCON, K. P. GUMMADI, P. DRUSCHEL, AND B. BHATTACHARJEE, *Measurement and analysis of online social networks*, in IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, New York, NY, USA, 2007, ACM, pp. 29–42.
- [43] B. A. NARDI, S. WHITTAKER, E. ISAACS, M. CREECH, J. JOHNSON, AND J. HAINSWORTH, *Integrating communication and information through contactmap*, Commun. ACM, 45 (2002), pp. 89–95.
- [44] J. NIELSEN, *Usability Engineering*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

- [45] J. NIELSEN AND R. MOLICH, *Heuristic evaluation of user interfaces*, in CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, 1990, ACM, pp. 249–256.
- [46] M. NOTTINGHAM AND R. SAYRE, *Rfc4287 - the atom syndication format*, December 2005.
- [47] M. OGRINZ, *Mashup Patterns: Designs and Examples for the Modern Enterprise*, Addison-Wesley Professional, 2009.
- [48] T. O'REILLY, *O'reilly network: What is web 2.0*, 2005. Retrieved at: 07.05.2009.
- [49] C. A. PETRI, *Kommunikation mit Automaten*, PhD thesis, Institut für instrumentelle Mathematik, Bonn, 1962.
- [50] A. RODRIGUEZ, *Restful web services: The basics*, IBM developerWorks, (2008).
- [51] F. SCHÖNEFELD, *Praxisleitfaden Enterprise 2.0 : Wettbewerbsfähigkeit durch neue Formen der Zusammenarbeit, Kundenbindung und Innovation ; Basiswissen zum erfolgreichen Einsatz von Web 2.0-Technologien*, 2009.
- [52] E. STEIN, *Organization memory: Review of concepts and recommendations for management*, International Journal of Information Management, 15 (1995), pp. 17–32.
- [53] D. M. STERLING, *Microsoft Office SharePoint Server 2007: The Complete Reference*, Mcgraw-Hill Professional, 2007.
- [54] H. TELLIOĞLU, *Cooperative work*, 2007.
- [55] —, *Keeping artifacts alive: towards a knowledge management system*, in CompSysTech '09: Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing, New York, NY, USA, 2009, ACM, pp. 1–6.
- [56] —, *Knowledge management with snapshots*, in PRO-VE, 2009, pp. 293–300.
- [57] M. TEPPER, *The rise of social software*, netWorker, 7 (2003), pp. 18–23.

- [58] R. TIDROW, C. BUECHLER, B. FOX, M. GORDON, J. MEDERO, N. MEHTA, J. POELMANS, P. PRUSSAK, AND C. REGAN, *Sharepoint 2007: The Definitive Guide*, O'Reilly Media, Inc., 2007.
- [59] W. M. P. VAN DER AALST, A. H. M. T. HOFSTEDÉ, AND M. WESKE, *Business process management: a survey*, in BPM'03: Proceedings of the 2003 international conference on Business process management, Berlin, Heidelberg, 2003, Springer-Verlag, pp. 1–12.
- [60] A. VAN KESTEREN, *Xmlhttprequest level 2 - w3c working draft*, September 2008.
- [61] H. WEIGAND AND M. LIND, *On the pragmatics of network communication*, in ICPW '08: Proceedings of the 3rd International Conference on the Pragmatic Web, New York, NY, USA, 2008, ACM, pp. 49–57.
- [62] A. WEISS, *Webos: say goodbye to desktop applications*, netWorker, 9 (2005), pp. 18–26.
- [63] E. WENGER, R. MCDERMOTT, AND W. M. SNYDER, *Cultivating Communities of Practice*, Harvard Business School Press, 1 ed., March 2002.
- [64] K. M. WIIG, *Knowledge management: Where did it come from and where will it go?*, Expert Systems with Applications, 13 (1997), pp. 1–14.
- [65] WIKIPEDIA, *Computer supported cooperative work*, July 2009. Retrieved at: 31.07.2009.
- [66] —, *Social network service*, August 2009. Retrieved at: 02.08.2009.
- [67] —, *Soziale software*, August 2009. Retrieved at: 02.08.2009.
- [68] —, *Exo platform*, February 2010. Retrieved at: 24.02.2010.
- [69] —, *Web application framework*, February 2010. Retrieved at: 13.02.2010.
- [70] V. WILLIAMS, *Microsoft sharepoint 2007 for dummies®*, John Wiley & Sons, Inc., New York, NY, USA, 2007.
- [71] B. WILLMS AND S. STAMER, *Enterprise 2.0 - Die Kunst, loszulassen*, Rhombos, 2008.
- [72] D. WINER, *Rss 2.0 specification*, March 2009.

Index

- A
 - AJAX 34, 66
 - Apache Shindig 73
 - Atom 12
- C
 - collective knowledge 8
 - communities of practice 36
 - contextual inquiry 30
 - CSCW 4, 35
- E
 - Enterprise 2.0 17, 67
 - enterprise mashups 14, 22
- F
 - feeds 31
 - FLATNESSES 19, 67
- G
 - Google Gadgets 39
 - groupware 4, 36
- I
 - iWiW 44
- J
 - JavaScript 34, 39
- K
 - knowledge management 8, 36
- L
 - Linked Data 75
- M
 - mashups 11, 14, 66
- N
 - network effect 36
- O
 - online communities 10, 35
 - organisational memory 9
 - orkut 45, 53, 57
- P
 - Portlets 73
- R
 - Resource Description Framework .. 75
 - RSS 12
- S
 - SaaS 15
 - semantic mashups 75
 - Sharepoint 20
 - SLATES 18, 68
 - SOA 15
 - social capital 37
 - social network platforms 44, 67
 - social networking 13
 - social software 12, 36
- U
 - user centered design 29
- W
 - Web 2.0 10, 34, 75
 - Web 3.0 75
 - web scraping 12

WebOS	29, 69, 72
Widgets	12, 73
X	
XML	34
XMLHttpRequest	34