ACTIVE
FP7-ICT-2007-1
Contract no.: 215040
www.active-project.eu

ACTIVE

Deliverable 1.3.3

Collaborative articulation of enterprise knowledge
Fully functional prototype and evaluation

Editor:	Benedikt Kämpgen, Karlsruhe Institute of Technology (KIT)
Deliverable nature:	Prototype (P)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	1st March 2011
Actual delivery date:	1st March 2011
Suggested readers:	Semantic MediaWiki community
Version:	1.0
Total number of pages:	34
Keywords:	enterprise knowledge structures, knowledge articulation, Semantic MediaWiki, lessons learned, case study

Abstract

For Task 1.3 in year three of the ACTIVE project, we describe our work towards an artifact that is enterprise-ready for collaborative articulation of knowledge. We evaluate this solution in a concrete scenario and give recommendations for how to apply it in other knowledge-intensive and collaborative use cases. We focus on enterprise knowledge in the form of information sources that employees need for their daily work. Also we built our work on Semantic MediaWiki as a representative semantic wiki which we intended to make palatable for the enterprise context. To assure our developments are applicable and suitable for enterprises we tailored our work to the case study partners, and together approached problems they encountered when deploying our solution. The technical challenges were fulfilled. Our evaluation is limited with respect to the non-functional requirements. Yet, we believe in case of a deliberate trade-off between our solution and scenario-specific needs in performance, maintenance efforts and usability our results help enterprises to release the potential of their employees' knowledge not only in our specific example but generally in crucial enterprise scenarios.
[End of abstract]

Impressum

[Full project title] ACTIVE – Knowledge-Powered Enterprise

[Short project title] ACTIVE

[Number and title of work-package] WP1: Enterprise Knowledge Structures

[Document title] Collaborative articulation of enterprise knowledge – Fully functional prototype and Evaluation

[Editor: Name, company] Benedikt Kämpgen, Karlsruhe Institute of Technology (KIT)

[Work-package leader: Name, company] Basil Ell, Karlsruhe Institute of Technology (KIT)

[Estimation of PM spent on the Deliverable]

Executive summary

In this document we describe our work for Task 1.3 *Collaborative Articulation of Enterprise Knowledge* during the third year of the ACTIVE project. The outcome is an artifact that can improve knowledge-intensive and collaborative activities in the enterprise context.

We focus on three aspects. First, articulation of knowledge in the form of enterprise knowledge structures. These are sources that provide employees with necessary knowledge for their daily work, e.g., customer descriptions, product specifications, and directories of employees, and which are not to be replaced but complemented by the artifact. Second, for such an artifact we built on semantic wiki functionality as represented by Semantic MediaWiki (SMW). Assumed in theory that it is suitable for the enterprise setting, we further developed it towards this goal. To assure our developments are applicable and suitable for enterprises, we have tailored our work of year three to our case study partners, and together acted upon challenges they encountered.

For instance, for BT's wiki we pulled together data from various sources, partly copied for being refined, partly displayed for being commented, only. To increase exploitation of the wiki content, e.g. for linking the data sources or for quality management, we frame it with an ontology. In the wiki faceted and keyword-based search allows for user-friendly querying. For Cadence' wiki we added support of discussions and controlled interactions between collaborating users, including contextualized RSS feeds. Accenture's wiki involved integration with available tools such as LDAP Active directory for convenient login or Microsoft Office for direct access to the wiki content.

As shown for proposal development, which we think is a representative scenario for knowledge articulation, we can say that SMW and its extension family fulfils the technical requirements to be beneficial in an enterprise context. Even though our evaluation is limited with respect to the non-functional requirements, we believe that in case of a deliberate trade-off between our solution's functionalities and scenario-specific needs in performance, maintenance efforts and usability our solution is generally applicable. Aligned along the technical challenges we resolved we give general recommendations of how to apply our solution in other typical enterprise scenarios:

- The enterprise needs to decide, what and how information is input. SMW supports form-based input of enterprise knowledge structures, formalized in an ontology. In proposal development, for instance, it can accelerate standard procedures, increasing the probability for a successful sale.
- It is to decide which information the user can easily request from the wiki, and in which form. Knowledge retrieval within SMW is available through faceted and keyword-based search as well as comprehensible views on content elements, e.g., discussion comments. In our scenario, it can make available additional information about a proposal to be consumed by novice team members.
- The enterprise should consider all relevant external knowledge sources for integration. Such external knowledge structures can be integrated in two ways: One in which the data should be copied into the wiki and made available to changes, and one in which the data should be queried and only be accessible for comments and discussions. If deployed for our scenario, the most relevant information about proposals is accessible directly from the wiki, e.g., product specifications.
- Quality check should be deployed. Data quality within the wiki is made transparent through context-aware views on the data. Using an enterprise ontology with data from the wiki, data quality problems can be identified semi-automatically, though not resolved automatically. For proposal development, it can help avoiding errors in proposals and reduce reviewing time.
- And also, external systems should be considered for integration. Because, SMW is all the more not an isolated data silo; enterprise knowledge structures in SMW can be used in combination with typical enterprise applications and infrastructure such as the LDAP Active directory, Microsoft Office, and RSS feeds. In our scenario, the proposal team can compose the actual proposal document more efficiently through an interplay between the wiki and Microsoft Word.

If opportunities are taken to integrate an increasing amount of machine readable data, at the same time keeping data quality high, and not replacing but enhancing traditional enterprise-wide tools, we expect Semantic MediaWiki, its extensions, and also recommendations of how to apply them to become best practices in enterprises to make use of their employees' knowledge in important business activities.

List of authors

Company	Author
Karlsruhe Institute of Technology (KIT)	Benedikt Kämpgen
Karlsruhe Institute of Technology (KIT)	Basil Ell

Contents

Abbreviations

Abbreviation	Description
SMW	Semantic MediaWiki
BT	British Telecommunications plc.
Cadence	Cadence Design Systems
OWL	Web Ontology Language
ETL	Extract, Transform, Load
RDF	Resource Description Framework
SPARQL	Simple Protocol and RDF Query Language
AKWS	ACTIVE Knowledge Work Space

Definitions

Term	Definition
Enterprise knowledge structures	Enterprise knowledge structures we define as the collection of various information sources in an enterprise that employees use to manage the knowledge for their daily work. Examples include customer descriptions, product specifications, and directories of employees.

1 Introduction

In this document we present the results that we achieved for Task 1.3 *Collaborative Articulation of Enterprise Knowledge* during the third year of the ACTIVE project. We have created a business-ready artifact for collaborative articulation of knowledge as it is crucial in many enterprise activities. Our work has been concentrating on three aspects:

Enterprise knowledge structures We aim at helping workers to collaboratively articulate enterprise knowledge. For that, we built artifacts that support knowledge workers to record, share and refine on relevant *enterprise knowledge structures*. Those structures comprise various information sources in an enterprise that employees use to manage the knowledge for their daily work. Examples include personal information sources such as email and instant messengers, enterprise internal sources such as task descriptions, product specifications, and directories of employees, as well as external information sources such as customer websites, blog entries, and Twitter feeds, all of which are not to be replaced but enhanced by the developed artifacts.

Deliverable 1.3.1 and also Paul Warren et al. [Warren et al., 2009] presented scenarios to be supported by ACTIVE technologies. Those scenarios have in common knowledge-intensive and collaborative activities such as proposal development. Employees often rely on their memory, previous experiences, and expertise to carry out those activities, which makes it difficult to share and find knowledge in the company. Traditional information management tools are not flexible or powerful enough to capture and allow collaboration around this knowledge [Simperl et al., 2010]. For instance, text editors do not allow monitoring of changes, distributed information in spread sheets is difficult to cross-link, relational databases are inflexible regarding modifications to their underlying schema, and common project management suites lack collaboration and discussion functionalities. Therefore, knowledge management environments are needed to complement traditional tools.

Semantic MediaWiki For managing enterprise knowledge structures, we have chosen to primarily use wiki technology that is enhanced with semantic technologies.

In two ACTIVE deliverables – Deliverable 1.3.1 [Krötzsch et al., 2009] and Deliverable 1.3.2 [Krötzsch et al., 2010] – we have presented various tools for overcoming challenges of dealing with enterprise knowledge. For instance, OntoGame [Siorpaes and Hepp, 2008] was introduced as an innovative interaction paradigm for collecting metadata in a goal-oriented and collaborative way. Another artifact presented for collaborative articulation of enterprise knowledge was *LiveNetLife*, which enables real-time communication between knowledge workers when accessing online resources. Those approaches were supplemented by functionalities semantic wikis offer. After year two of the ACTIVE project, we have presented *OntologyEditor*, a wiki-based artifact for collaborative enterprise modelling and repair. Wikis in general have long found their way from the Web 2.0 into the intranets of companies of all sizes [Drakos et al., 2009]. Just as wikis in the public web, these so-called enterprise wikis provide their advantages of low usage-barriers and direct benefits within a company intranet.

In Deliverable 1.3.1 [Krötzsch et al., 2009] we have proposed Semantic MediaWiki (SMW) as a representative semantic wiki to be applied to enterprise contexts. Yet, from the beginning, we have anticipated challenges that would hinder the semantic-wiki-based collaborative knowledge articulation within a company: we intended to achieve *ease of use* in an enterprise wiki by concepts such as form-based editing, what-you-see-is-what-you-get editing, and intuitive annotation interfaces; for accessing the knowledge we have suggested faceted browsing and process visualization; the challenge of seamless *desktop knowledge exchange* we would overcome with widely used export formats and desktop-based writing access to SMW; the expected need for *higher-level knowledge structures* in SMW we have fulfilled through ontology building and repairing tools; and also, we have taken into account the problem of *access control* such as group-wise read and write restrictions. We have selected, supported, and implemented several extensions to SMW in order to make it palatable for enterprises.

Case study partners To assure applicability and suitability for enterprises, we have tailored our work in year three for Task 1.3 to the case studies. More concretely, we have directed our development and maintenance of an ACTIVE tool for knowledge articulation to the needs of our case study partners British Telecom (BT), Cadence Design Systems (Cadence), and Accenture.

Our solution fulfils the technical requirements of our case study partners. In a specific scenario we show its advantages, and then draw lessons learned of how to apply it to other knowledge-intensive and collaborative activities. The outcome of Task 1.3 we present here consists of SMW in its current status, its generic and specific extensions, and recommendations for their usage in real world enterprise scenarios.

The remainder of this deliverable is structured as follows. In Section 2 we summarize our work done in year three and describe how it is related to ACTIVE results and each case study. In Section 3 we evaluate our work by analysing the current status of SMW for dealing with enterprise knowledge structures in a concrete use case. Then, in Section 4 we describe lessons learned that shall help enterprises to apply our solution in other enterprise scenarios, after which, in Section 5, we summarize the outcome of Task 1.3, and describe possible future work.

2 Work in year three – SMW tailored to the case studies

In the following, we will give a summary of the work we have done for each of the case studies. Note, we only describe those extensions that were newly introduced or considerably modified in year three. In Appendix A.1 we have added a complete list of extensions that we have used within the case studies.

British Telecom

In the form of a BT proposal wiki, SMW at British Telecom focuses on the knowledge transfer and information needs of sales people at BT Business, namely technical consultants, solution consultants and sales specialists. As an example, BT is encouraging experts to answer questions for a team working on a proposal.

Together with BT, we have built on ontologies presented in Deliverable 1.1.2 [Ell et al., 2010] and refined and integrated them to an ACTIVE Upper Level Enterprise Ontology to formalize the enterprise knowledge structure created in the wiki. This ontology describes elements that are relevant for typical enterprise scenarios – e.g., employees, customers, and products – as well as elements that are important for our specific proposal scenario, e.g., proposals and their assigned team members. If filled with instance data from the wiki the ontology would improve search and browsing of enterprise information, e.g., proposals and their assigned team members. For this upper level ontology we combined two kinds of ontologies:

Ontologies describing content in BT proposal wiki Concepts and properties that are deployed in BT proposal wiki, categories such as people, customers, and products, or properties such as customer visions, included products, and priorities of customer issues. They are motivated by data that wiki users manually create, and also from data available for integration, e.g., people directories, proposal documentations and product descriptions. Also included here is the concept of customer issues, around which collaboration within BT proposal wiki is centred. In SMW, the expressivity on the terminology level is deliberately restricted to concepts and properties, subconcepts and subproperties, and equivalence relations, which is why this part of the ontology is modelled on such an expressivity level.

Ontologies of external data sources Concepts and properties that are used in other datasets. For instance, we have linked elements from the wiki to concepts in FOAF[1], GoodRelations[2], and dbpedia[3]. If exported as RDF, the data in the wiki can be added to the ontology and then be integrated with other datasets, e.g., countries from dbpedia. Such data can provide means for semantic search and quality checks through reasoning. Later in this section we will describe our efforts of manually linking datasets within BT; for linking instances in the wiki to instances in external datasets even more manual effort is needed as those externally created datasources are more likely to contain inconsistencies and errors. Also, as it is not planned to represent this part of the ontology in SMW, the ontologies were not restricted in expressivity. In fact, the combined expressivity of the ontology according to Protégé is $\mathcal{SHOIQ}(\mathcal{D})$ which is above the expressivity of OWL DL ($\mathcal{SHOIN}(\mathcal{D})$) and below the expressivity of OWL 2 DL ($\mathcal{SROIQ}(\mathcal{D})$).

[1] http://xmlns.com/foaf/spec/

[2] http://www.heppnetz.de/ontologies/goodrelations/v1

[3] http://dbpedia.org/About

We partly used the SMW OntologyEditor [Krötzsch et al., 2010] for creating the ontology, however, as it was not a highly collaborative effort to create the ontology and participants were technically skilled, we mainly used Protégé[4] and the NeOn Toolkit[5].

For seeding the wiki with instance-level content, we implemented a case study specific extension *BT-CSV-Import*[6] that allows to import tabular data as wiki pages assigned to a certain category and semantically described with properties. For instance, we use it to integrate proposal data stored in a Microsoft Access database. For security reasons we decided to export the data as CSV data instead of directly accessing the database; for practical reasons we run the export on a fixed schedule and not every time a change happens to the data. The data contains columns such as for the reference number, for included products, for the customer name, and for the proposal status, and populates properties hasRefNo, hasProduct, hasCustomer, and hasStatus of proposal wiki pages with certain reference numbers. People data is directly integrated in user pages. Those pages for each registered user are normally created after a user for the first time logs onto the wiki; in this scenario, however, they are created beforehand. In case of changes to the underlying CSV data, existing proposal pages are updated without changing comments and other manually added annotations. Since comments might then be inconsistent to this new information, it is clearly indicated to the users when data sources have last been updated and comments changed.

As the functionality of BT-CSV-Import is limited for most use cases, we helped extending it into a new extension *RDFIO*[7] to also import general RDF. RDFIO works as a semantic storage integration; it provides a SPARQL endpoint for the wiki. Using this standard query interface, data from the wiki can be queried by external SPARQL clients such as Twinkle[8], moreover, structured content can be modified using SPARQL Updates. Part of this external data, available as CSV files, and integrated into BT proposal wiki, is not supposed to be changed by users, but only commented. For that, we developed an SMW *Mediator* extension. It allows to query the data directly from the CSV file, instead of accessibly storing all imported data in the wiki. Mediator allows to have such *External Queries* not only of CSV data but generally of structured data sources, e.g., Freebase. Mediator is subject of Deliverable 1.4.3 [Ell et al., 2011]. To let CSV-Import and RDFIO directly access and modify structured data within the wiki, we used the extension *SMWWriter*[9] [Krötzsch et al., 2010].

Besides better search by semantic annotations, a further reason to import data from various sources, such as people directories or product specifications, is it to link those data. For instance, we link proposals to person data of the development team. For that, data cleansing and pre-processing were needed, e.g., replacing of common spelling errors, acronyms, and abbreviations. Tools such as Google Refine[10] can help manually identifying and resolving integration issues. In BT, BT-CSV-Import is used in combination with an upstream ETL pipeline that retrieves and pre-processes proposal, person, and product data before importing it into the wiki. More information about this can be found in Deliverable 9.2.3 [Thurlow and Gu, 2011].

For information retrieval in BT proposal wiki we deployed *Semantic Result Formats*[11] extension, especially its integrated faceted search visualization *Exhibit* to provide overviews over the imported data. For instance, proposals can be searched and browsed by status and product, and people by their role. Exhibit's visualisation is limited in the sense that it cuts down huge lists to around 500 result entities. We found a way to deal with this issue; instead of querying for all items at the same time, the user drills down to smaller subsets of the query result, e.g., proposals within certain years or months. For "contagging and contextualized search" as introduced in Deliverable 3.1.2 [Bloehdorn et al., 2010], we deployed the corresponding extension that connects BT proposal wiki to the ACTIVE Knowledge Work Space (AKWS) [Dolinsek et al., 2010]. Inside BT proposal wiki, context is defined as a group of information resources, namely wiki pages. When modified, wiki pages are bound to the current context, to be highlighted and more easily found when users work in this context. Processes from the AKWS can be stored in SMW and then be modified there. For that, we developed an XML export, that allows to rewrite the modified processes to the AKWS.

As mentioned above, collaboration in BT proposal wiki is centered around customer issues. We have

[4] http://protege.stanford.edu/
[5] http://neon-toolkit.org/wiki/Main_Page
[6] This extension is highly case-study-specific and therefore will not be published.
[7] http://www.mediawiki.org/wiki/Extension:RDFIO
[8] http://www.ldodds.com/projects/twinkle/
[9] http://semantic-mediawiki.org/wiki/Help:SMWWriter
[10] http://code.google.com/p/google-refine/
[11] http://semantic-mediawiki.org/wiki/Help:Semantic_Result_Formats

created an interface of semantic queries, templates and forms for group discussions and community questions about these. BT sales people and technical specialists want to be notified about high-priority customer issues. Therefore, we configured a suitable RSS feed that is supplied with information from a semantic query on proposal data within the wiki and which provides links to the relevant page. BT employees can subscribe to this feed and are informed about important issues without having to visit the wiki.

Deliverable 9.2.2 [Thurlow, 2010] has identified several usability issues around Semantic MediaWiki which we have worked on in year three. Some, found crucial by potential users, we could help to resolve, e.g., providing an cleaned up navigational structure and comprehensible help texts; having edit pages looking similar to rendered pages; and a more prominently placed search function. Also, response-time of AJAX autocompletion feature – used for typing in cross-selling products – has improved. As an extension to MediaWiki[12] SMW cannot overcome all usability issues. For instance, the procedure of first uploading a file and then linking it from a wiki page is inherently cumbersome, but could only slightly be improved by using forms for uploading. Similar is valid for editing of wiki pages. Wikimedia has started a Usability Initiative[13] that intends to improve the usability. We tested their prototypical usability extension[14] and found it not sufficiently stable for full integration.

Cadence Design Systems

SMW at Cadence shall help electronics design engineers as they navigate and instantiate complex design processes, and, for instance, share knowledge with development teams.

In Cadence wiki the openly available PSI upper-level ontology [Tilly, 2010] is used. In year three, there has been an update from version 2.0 to version 2.3 of its reference specification. We modified the usage of the ontology inside the wiki to reflect this update.

One important piece of functionality is related to the support of discussions and controlled interactions between collaborating users. Therefore, we added a group discussion functionality. With it, discussions can be started for all types of knowledge that is inserted into the Cadence wiki, e.g., tasks, tools, activities, and design artifacts. Single discussion comments are displayed in context: on their wiki page, other comments related to the discussion topic are also listed. Participants of a discussion are alerted about changes to the discussion. Also, pro and contra arguments can be grouped and summary boxes shown to foster collaboration and efficient consensus making. We allow to explicitly invite users to discussions; users see invitations on their user page, and also can be notified by a personal invitation RSS feed. The discussion functionality is not based on an extension such as Liquid Threads[15] but can completely rely on *Semantic Forms*[16] and templates, which makes it more flexible for refinements. A more detailed description as well as screenshots of the discussion functionality can be found in Deliverable 1.4.3 [Ell et al., 2011]. We evaluated, whether more sophisticated user rights management with discussions could be realized. For instance, so that users may deactivate their own comments or administrators may discard comments, and so that only administrators may start a discussion. However, those user group restrictions would neither align with wiki philosophy of "everyone can read and write" nor would they have been able to implement reliably without great changes to the MediaWiki software.

For instance, regarding visualisation of processes using our Process Visualisation [Tilly et al., 2010] extension, discussions can be used to propose modifications to processes. Processes can be annotated and then queried and browsed. For that, we created certain wiki pages that use semantic templates to offer views on processes. For instance, in order to give a measure about process execution, iterations of tasks are counted and displayed together with a process. Besides those predefined queries, we also provide keyword search functionality. Instead of *Ask The Wiki* extension which revealed limitations, we deployed our new light-weight version *AskQ* extension. For usability reasons, expressivity of those queries are restricted to retrieve only tree-structured information. Process Visualisation and keyword search in SMW are described in more detail and with screenshots in Deliverable 3.1.3 [Kämpgen et al., 2011a]. Also, we implemented an SMW extension *Semantic Project Management*[17] that allows to render project structures as Gantt charts and export them as XML

[12] http://www.mediawiki.org/wiki/MediaWiki
[13] http://usability.wikimedia.org/wiki/Wikipedia_Usability_Initiative
[14] http://www.mediawiki.org/wiki/Extension:UsabilityInitiative
[15] http://www.mediawiki.org/wiki/Extension:LiquidThreads
[16] http://www.mediawiki.org/wiki/Extension:Semantic_Forms
[17] http://www.mediawiki.org/wiki/Extension:Semantic_Project_Management

files for use in Microsoft Project. See Figure 1 for an example.

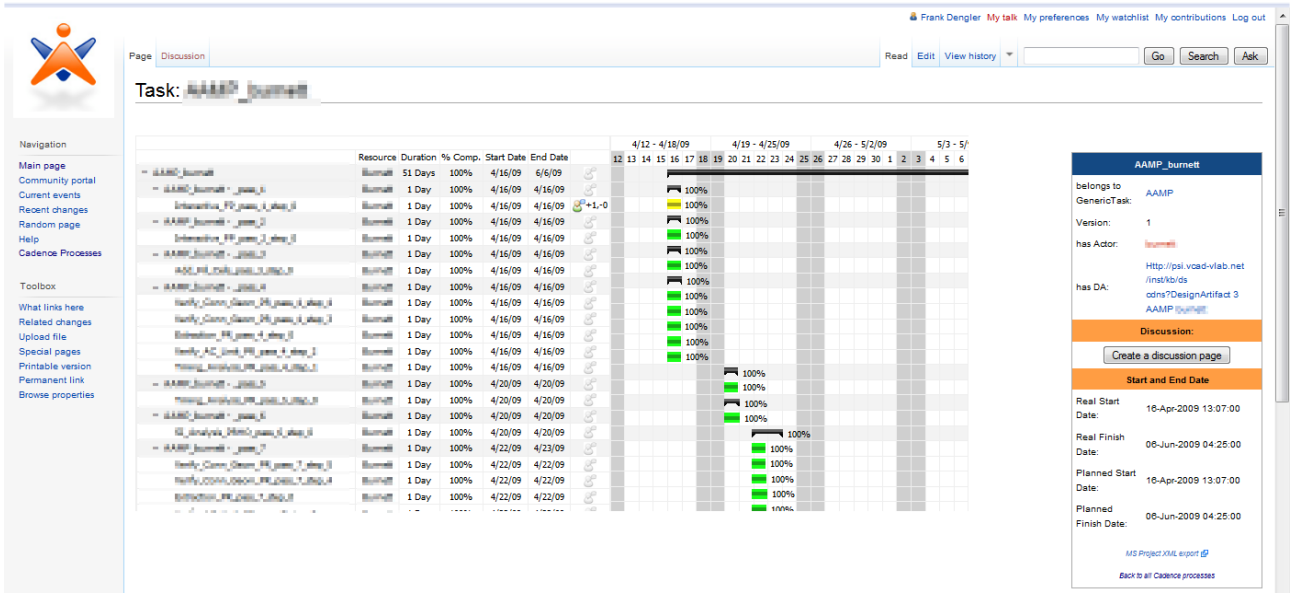

Figure 1: Gantt chart with Microsoft Project export

As described in Deliverable 10.2.2 [Ermolayev et al., 2010], for Cadence wiki, usability problems were encountered, similar to the ones identified in BT. We made user interface improvements such as showing labels instead of URIs and providing more help and documentation texts. Also, we improved search within Cadence wiki, e.g., by applying more intuitive query templates, or more use of Semantic Result Formats extension. Again, not all usability-related inquiries could be fulfilled entirely, due to MediaWiki backbone.

Accenture

With their proposal development wiki, Accenture intends to achieve more effective knowledge management and thereby to improve the sharing and reuse of information that is contained within the enterprise. This, without replacing but rather seamlessly integrating with existing knowledge repositories.

Similar to BT, in Accenture wiki, a case-study-specific ontology is deployed. Djordjevic et al. [Djordjevic et al., 2010] give more information about this ontology.

Instead of having for each user of the wiki to create a separate account, we deployed the *LDAP Authentication*[18] extension to use the existing Active directory within Accenture.

Process visualization deployed at Cadence wiki has not fulfilled the needs of Accenture; in their proposal development wiki, processes are not only to be automatically inserted and then visualized for comments to the user but also to be inserted manually, in a user-friendly way. Therefore, we built a new Process Editor extension for SMW that comprises both a textual and a graphical editor for processes. It is presented in Deliverable 3.1.3 [Kämpgen et al., 2011a], together with an evaluation. PDF documents can be uploaded to SMW and are automatically tagged with keywords from an Accenture taxonomy. Each word in the document that is part of the taxonomy is transformed into a tag. Therefore an article becomes tagged with each word that is contained both in the article and in the taxonomy. This makes uploaded documents easier to find using semantic queries.

Exhibit's faceted browsing is not sufficiently efficient in some cases. In these cases, we created single pages of each property value, from where the user can browse to other elements featuring this value. For instance, on the page "Skill - Java Enterprise Edition" all people having this skill are listed. The collaborative proposal development workspace enables team members to find people by filtering on home office, projects or language skills. The SharePoint people directory web service was too slow to entirely import all 170 000 Accenture workers. It would have taken around 80 days to import pages of all Accenture employees into the SMW. Instead, only samples of the database are imported on demand; if the project manager instantiates a new proposal development workspace in the wiki he uses wiki syntax, e.g., $\{\#PortalImport : scope =$

[18] http://www.mediawiki.org/wiki/Extension:LDAP_Authentication

$People||name = XXXX\}$, to indicate people information to be imported [Fullarton, 2010]. Loading times of SMW are not influenced by those performance issues of external systems. In general, SMW showed sufficiently fast, and did not require additional caching mechanisms, e.g., Squid[19].

We deployed WikiTags extension [Bloehdorn et al., 2010] for Accenture wiki. When composing the proposal with Microsoft Word, users can insert structured information from the wiki related to a certain term directly into an office document. Further development of WikiTags extension is done by Vulcan. Also, we improved RDF export of SMW. Consistency checking, done for Accenture as described in Deliverable 1.4.3 [Ell et al., 2011] can be used to check facts entered into an Office document against facts retrieved from SMW.

Trials and feedback from potential users also revealed the problem that non-technicians find the user interface of MediaWiki and Semantic MediaWiki not sufficiently user-friendly. We improved the usability by using *Halo*[20] extension, which also provided a professional look for the wiki.

3 Evaluation – Analysis of SMW

In this section, we foster our claim that SMW in its current status combined with generic and specific extensions generally can bring benefits for knowledge articulation in knowledge-intensive, collaborative enterprise scenarios. We align our evaluation along the scenario of proposal development. Proposal writing has been less relevant for Cadence Design Systems, and mainly been relevant for BT and Accenture. Still, we regard it as a representative example for articulation of enterprise knowledge, since it involves many knowledge-intensive and collaborative activities:

Traditionally, proposal development starts in response to a client's request for information (RFI) or invitation to tender (ITT). The proposal manager selects a suitable team and monitors and documents the development progress, e.g., using a Microsoft Access database. The team gathers information from internal (e.g., product specifications) and external (e.g., customer websites) information sources, and discusses possible content for the proposal in meetings, through phone and per email. For more complex customer issues also technical specialists are consulted. The proposal document is eventually composed within a Microsoft Word document. After a thorough review, it is sent to the customer.

In the following, we shortly summarize the results of our evaluation and give additional information: Proposal development imposes functional requirements regarding knowledge 1) input and 2) retrieval; integration of 3) external data sources and 4) external tools; and 5) quality management, as well as non-functional requirements regarding 6) performance, 7) maintenance efforts, and 8) usability. According to the case study partners our solution fulfils the functional requirements. More importantly, if tested in a trial, the success of each functionality should be measurable, as exemplified for proposal development in the following:

Creating proposal development information Just as writing a letter requires the address and stamp at certain locations, a predefined structure helps people to follow certain procedures. Thus, one measurement would be the time a proposal manager needs to input necessary information for a proposal. In our solution input is done through forms, and uses semantic information. For instance, when inserting team members, autocompletion suggests only people. Similarly, collaboration should be more efficient through the wiki as it can be based on previous discussions, done asynchronously, also involving the wider team of technical specialists. Therefore, also the time it takes for team members to finish information gathering and discussion could be reduced. In general, the entire time needed to finish a proposal using the wiki could be compared with traditional development cycles. As with most business processes, the quicker the proposal development the less expensive it is. Also, a customer may have more trust in BT if the proposal is delivered faster [Thurlow and Gu, 2011].

Retrieving proposal information An office document typically stands for itself, therefore most input fields contain just simple alphanumeric values. The team members need to find meta information about these values by themselves. In a wiki, instead, those fields can be filled in with terms described by other pages in the wiki. Links from the proposal to other information offer relevant additional information sources about a proposal.

[19] http://www.squid-cache.org/

[20] http://www.mediawiki.org/wiki/Extension:Halo_Extension

Similarly, overviews and search of proposal information should make it easier for new employees to comprehend finished proposals without consuming expensive time from an expert. Time and effort it takes before an employee can contribute to proposal development would be another measure for success.

Integrating proposal-related information Assuming the wiki is accessible by all employees, all information open to them is potentially interesting. This includes public information such as blog entries, news items, and Twitter feeds, as well as enterprise internal information such as task descriptions, the menu of the canteen, and the company organigram. For proposal development, most relevant information sources can be identified, e.g., finished proposals, person data, and product specifications. These information sources are not to be replaced but complemented by the wiki which works as a hub for getting familiar with, commenting on, and linking between those sources, which in the end should lead to more mature information in proposals. Therefore, high usage of this integrated data sources by the users would be another success indicator, which could be measured by page views or line of comments.

Improving proposal data quality Proposals that are composed by putting together unstructured information, especially if done collaboratively, are open to inconsistent or wrong information. Consequently, proposals are extensively reviewed. In the case of the wiki, as long as the information resides in there in the form of structured data it helps to make data quality transparent. For instance, a wrong customer name refers to a non-existing page and is visible as a red link, and missing information is easily identified in sparse queries. After an export, inconsistencies in the data can be found, e.g., if no date can be matched to the due date or instead of an organisation, a product is given as a proposal customer. Those issues can then be easily resolved. Reviewing the proposal after it is composed with content from the wiki still needs reviewing, however, we expect it to reveal on average fewer errors and inconsistencies and consume less time from the reviewer as in the non-wiki way.

Interplay with tools for proposal development Traditional enterprise tools for proposal development should be complemented by the wiki, as it is done for Microsoft Word. If used in combination with the wiki, information for the proposal can directly be retrieved from the office software, resulting in fewer copy-and-paste actions and switching to other software such as the browser. Again, this would lead to faster development. Therefore, the time spent and data transferred for using both systems would be another success indicator.

More information about our evaluation work we have published at a workshop of a knowledge management conference in February 2011 [Kämpgen et al., 2011b]. The paper we have added to Appendix A.2. Our hypothesis we discuss in this paper is that Web 2.0 and Semantic Web technologies complement each other not only in the World Wide Web[Ankolekar et al., 2007] but also for crucial activities within the boundaries of company intranets. We explain how Semantic MediaWiki can resolve technical challenges and provide opportunities in the enterprise context and then evaluate our findings by explaining how this solution can be deployed for the representative knowledge-intensive and collaborative activity of proposal development.

After our internal evaluation, trials have been started in each case study for evaluating the solution in a realistic environment; their results will be described in deliverables of BT (Deliverable 9.2.3), Cadence (Deliverable 10.2.3), and Accenture (Deliverable 8.2.3). However, we do not expect the trials to be able to fully measure the success of our solution, because when we deployed it at our case study partners the systems showed limitations in regard to the non-functional aspects:

Performance Importing large amounts of data was not always feasible, e.g., due to the web service of the people directory importing all 170.000 Accenture employees would have taken 80 days.

Maintenance efforts With semantic technologies, we hoped to decrease efforts needed to deploy and run the systems. However, much work was needed to instantiate a professional system. Modelling and implementing the ontology; updating interfaces; implementing batch scripts for loading data from external sources; creating views on the data; and offering forms mainly was manual work. Nevertheless, on the long run, we expect it to be less work than in traditional wikis, due to its flexibility and control over the data, especially if it is grounded to an explicit ontology, and even if modifications to the underlying ontology occur.

Usability Also, we did not expect usability of MediaWiki being an issue to such an extent. MediaWiki is used on Wikipedia and therefore millions of people use it, and ten thousands of people actively collaborate with it. Still, this interface was not ideal for the common employee wanting to use the wiki for his daily work. For instance, users had problems with wiki text editing; data upload; and navigation. Semantic technologies do not per se increase usability. Semantic forms or Ontoprise's Halo extension did help, but SMW could not compete with user interfaces provided by Microsoft SharePoint, for instance. User training and related expenses can improve the situation, still, usability issues keep to be a major barrier of using an enterprise wiki, be it semantic or not. Integration of Semantic MediaWiki functionality into SharePoint, as pursued by Ontoprise[21] seems a promising alternative.

It would have been possible to eventually overcome these non-functional requirements, however, as we expected them to be very case-study-specific and their solutions not generally applicable we put more emphasis on the functional part of our solution; those requirements we fulfilled. If a trade-off between our solution's functionality and scenario-specific needs regarding performance, maintenance efforts and usability is found, our solution could show real business advantages. More than that, as we regard proposal development as a representative example for knowledge-intensive and collaborative activities, we believe that our solution can generally address technical challenges of enterprise knowledge articulation scenarios. In the next section we will describe how we would recommend enterprises to proceed in a new scenario.

4 Lessons learned

In this section we give some recommendations for applying our results in a concrete enterprise scenario. The technical challenges we solved each give a specific aspect to decide and act on. The resulting guidelines are not to be processed consecutively but rather iteratively, agile on changes to the enterprise requirements. All of these aspects should be considered with a good understanding of the target audience and the aim of the wiki, though those might be evolving over time. Every decision may imply to use publicly available extensions to MediaWiki or Semantic MediaWiki or to build customary ones.

Creating structured information One important aspect is it to decide on the structured information that shall be input. Early deployment choices can have profound downstream impact [Grudin and Poole, 2010]; also, in our case, an initial ontology needs to be created and applied to the wiki but it does not need to be final, rather should be modified and refined continuously. Only structured information can be retrieved or checked on errors, however imposing more structure on the input should be decided with respect to maintenance efforts for updating the ontology and the wiki, and usability of the way information is input.

Retrieving information Also, it is to decide, what information users can easily retrieve from the wiki, and in which form. Not only overviews should be given, the information made browseable and searchable through the system but also it should be possible to easily take a snapshot of current information, e.g. for distribution to a customer [Grudin and Poole, 2010]. Again, usability is an important point, as well as performance of queries.

Integrating external information Generally, all relevant external data sources should be considered for integration. Traditional wikis lack the possibility to integrate various kinds of information, e.g., Visio diagrams [Grudin and Poole, 2010]. With SMW, even tight integration with such sources is possible. It needs to be decided, to what grade, what for, and how they should be integrated; most importantly, whether these sources are supposed to be modified or displayed only. These decisions depend on the characteristics of the integrated data sources. Efforts for the initial integration and for updating the interfaces need to be considered, e.g., before deploying a complex ETL pipeline. Performance and availability of external data sources is another important aspect to keep in mind. Relevance and features of external sources may be changing (e.g., in the case of Linked Open Data), their integration therefore should regularly be reconsidered.

Improving data quality For structured information it should be considered whether quality checks are possible and worth the maintenance effort. It is to decide what to do with found errors. Also, performance

[21] http://www.ontoprise.de/de/loesungen/semanticwiki-for-sharepoint/

might be a relevant factor, e.g., if reasoners are applied. A problem of traditional wiki is that information gets stale and is difficult to clean [Grudin and Poole, 2010]; quality checks can reduce efforts of so-called wiki gardeners.

Interplay with other enterprise tools Similar to the case of external information one needs to decide what external systems in the company should be accessing the wiki. Every tool that is used in the specific enterprise context might be useful to complement by the wiki, be it through direct HTTP access and an API, or after an export of wiki data. Integration should especially be considered in the case of familiar and highly-used communication systems such as instant messengers, emails, and SharePoint, which otherwise employees will resort to in the case of problems [Grudin and Poole, 2010]. The decision for integrating a tools depends on performance of the connection, maintenance efforts for updating necessary interfaces, and usability of the integration.

Or solution is not as light-weight as Twitter feeds or blogs and not as comprehensive as full-suite groupware sytems. Still, we are convinced that our results of Semantic MediaWiki, its extensions, and usage recommendations are flexible and powerful enough to be applied to many typical enterprise scenarios, be they limited-duration projects, long-term knowledge management deployments, or something in between.

5 Conclusion

In this section we summarize our work in year three for T1.3 *Collaborative Articulation of Enterprise Knowledge*, and discuss its outcome also with respect to possible future work: For managing enterprise knowledge structures using a semantic wiki our work was centred around technical challenges of knowledge input and retrieval, integration of external data sources and external tools, as well as quality management. In order to overcome those challenges we provide a solution made of Semantic MediaWiki, its extensions, and recommendations for their usage.

Wikis are already widely made available in enterprises, whether they are used successfully by the employees, however, depends on many aspects [Grudin and Poole, 2010]. Organizational and human success factors are crucial, yet, they can be influenced mainly by social behaviour, which much has been written about and we did not focus on. Rather more potential we anticipated in the technical challenges. Traditional wikis provide useful functionalities such as easy reading and writing, storage of text and media, and change tracking; still, their functionality is limited, e.g., with respect to control over their content. We anticipated to resolve the technical challenges using a semantically enhanced wiki, represented by Semantic MediaWiki.

Now, after three years of work, SMW supports guided input of enterprise knowledge structures, formalized in an enterprise ontology. Within SMW, the expressivity on the terminological level is deliberately restricted; most importantly, users can easily insert new individuals. If exported as RDF, the data in the wiki can be added to the enterprise ontology and then full expressivity of OWL 2 DL can be exploited. Knowledge retrieval within SMW is improved through faceted and keyword-based search and comprehensible views on content elements, e.g., discussion comments. Again, for usability reasons, expressivity of those queries are restricted to retrieve only tree-structured information. External knowledge structures can be integrated, for two scenarios: One in which the data should be copied into the wiki and made available to changes, and one in which the data should be queried and only accessible for comments and discussions. Data quality within the wiki is made transparent through context-aware views on the data. Grounding the data within the wiki to an external ontology, data quality problems can be identified semi-automatically, and then resolved manually. In addition, SMW is all the more not an isolated data silo; enterprise knowledge structures in SMW can be used with enterprise-wide tools such as the LDAP Active directory, Microsoft Office, and RSS feeds.

Evaluation revealed that our solution meets the functional requirements, however is limited by the non-functional requirements; those can be resolved if a trade-off between our solution's functionality and enterprise-specific needs regarding performance, maintenance efforts and usability is found. For future work, it would be interesting to more closely examine this necessary trade-off. Given such a trade-off our solution could show real business advantages. More than that, we believe that our solution is powerful and flexible enough to generally address technical challenges of enterprise knowledge articulation scenarios, and our lessons learned give guidelines how to generally apply our solution in new enterprise scenarios.

To have sustainable existence of SMW instances within enterprise ecosystems we see our solution as a starting point for several opportunities. First, with respect to integration of external information. We think that much data in today's enterprises are not taken full potential of. For instance, the trend of publishing data in structured form[22] offers possibilities to enhance and make better use of enterprises knowledge. However, on the one hand importing huge amounts of information into the wiki might be unfeasible due to availability or performance, might require great efforts of pre-processing and synchronisation, and might unnecessarily over-engineer the wiki content and complicate its browsing and searching. On the other hand only displaying the information might not have sufficient semantic implications to efficiently capture and allow collaboration about knowledge. Also, comments about displayed data might become invalid after an update. We have pursued a combination of partly importing and partly displaying external data, also making it transparent to users when data was updated and modified. Such a combined approach can be further elaborated: External enterprise knowledge structures would first be queried and displayed, then, after checked for validity and usefulness, and possibly curated by the users, would be imported partly into the wiki. An example of such an approach is Shortipedia[23], which allows to query linked data and import single facts into the wiki [Vrandecic et al., 2010]. Also, since the last user and developer meeting in September 2010, a closer integration of Semantic MediaWiki and external information is discussed in the SMW community[24].

Second, as the amount of data managed and stored inside wikis increases, so does the need for automatic quality checks. A wiki should be flexible and not restricting the users. Still, we think that a highly collaborative environment such as a wiki can only be successful on the long run if users and administrators are supported in keeping content quality high as it is possible with our solution.

Third, to not overwhelm users but incrementally introduce them to advantages SMW can provide, we especially see opportunities in enhancing traditional tools. Regarding this, we want to highlight Vulcan's plans with project Halo to make SMW more like an application platform [25].

If these endeavours are further pursued, we expect Semantic MediaWiki, its extensions, and usage recommendations to become best practices in enterprises to make use of their employees' knowledge in important business activities.

[22] http://linkeddata.org/
[23] http://shortipedia.org/index.php/Main_Page
[24] http://semantic-mediawiki.org/wiki/SPARQL_and_RDF_stores_for_SMW
[25] http://smwforum.ontoprise.com/smwforum/index.php/SemanticMinds_JesseWang

References

[Ankolekar et al., 2007] Ankolekar, A., Krötzsch, M., Tran, T., and Vrandečić, D. (2007). The two cultures: mashing up web 2.0 and the semantic web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 825–834, New York, NY, USA. ACM Press.

[Bloehdorn et al., 2010] Bloehdorn, S., Dengler, F., Vrandečić, D., and Tilly, M. (2010). Knowledge filters – demonstrator. Deliverable, ACTIVE.

[Djordjevic et al., 2010] Djordjevic, D., Ghani, R., and Fullarton, D. (2010). Process-centric enterprise workspace based on semantic wiki. In *International Conference on Knowledge Management and Information Sharing (KMIS 2010)*.

[Dolinsek et al., 2010] Dolinsek, I., Stajner, T., Moreno, C. R., and Tilly, M. (2010). Fully functional prototype of active knowledge workspace. Deliverable, ACTIVE.

[Drakos et al., 2009] Drakos, N., Rozwell, C., Bradley, A., and Mann, J. (2009). Magic quadrant for social software in the workplace. Gartner RAS Core Research Note G00171792, Gartner.

[Ell et al., 2010] Ell, B., Dengler, F., Djordjevic, D., Garzotto, F., Krötzsch, M., Simperl, E., Siorpaes, K., Vrandečić, D., and Wölger, S. (2010). Conceptual models for enterprise knowledge (final models and evaluation). Deliverable, ACTIVE.

[Ell et al., 2011] Ell, B., Kämpgen, B., Sorg, P., Vrandečić, D., and Kienzle, S. (2011). Knowledge leveraging fully functional prototype and evaluation. Deliverable, ACTIVE.

[Ermolayev et al., 2010] Ermolayev, V., Jentzsch, E., Ivasiva, A., Ivasiva, A., and Ivasiva, A. (2010). Demonstrator and validation report. Deliverable, ACTIVE.

[Fullarton, 2010] Fullarton, D. (2010). Enterprise 2.0: Using semantic mediawiki for proposal writing at accenture. Diploma thesis, Institute of Applied Informatics and Formal Description Methods - Karlsruhe Institute of Technology.

[Grudin and Poole, 2010] Grudin, J. and Poole, E. (2010). Wikis at work: success factors and challenges for sustainability of enterprise Wikis. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, pages 1–8. ACM.

[Kämpgen et al., 2011a] Kämpgen, B., Dengler, F., and Herzig, D. (2011a). Context-aware knowledge filters – fully functional prototype and evaluation. Deliverable, ACTIVE.

[Kämpgen et al., 2011b] Kämpgen, B., Ell, B., Simperl, E., Vrandečić, D., and Dengler, F. (2011b). Enterprise wikis: Technical challenges and opportunities. In *WM2011: 6th Conference on Professional Knowledge Management*, number 182 in Lecture Notes in Informatics (LNI), Bonn. Bonner Köllen Verlag.

[Krötzsch et al., 2010] Krötzsch, M., Bürger, T., Luger, M., Vrandečić, D., and Wölger, S. (2010). Collaborative articulation of enterprise knowledge (demonstrator). Deliverable, ACTIVE.

[Krötzsch et al., 2009] Krötzsch, M., Dengler, F., Siorpaes, K., and Vrandečić, D. (2009). Collaborative articulation of enterprise knowledge (early prototypes). Deliverable, ACTIVE. Available at `http://www.active-project.eu/fileadmin/public_documents/d131-final.pdf`.

[Simperl et al., 2010] Simperl, E., Thurlow, I., Warren, P., Dengler, F., Davies, J., Mladenic, D., Gomez-Perez, J. M., Moreno, C. R., and Grobelnik, M. (2010). Overcoming information overload in the enterprise: the active approach. *IEEE Internet Computing*.

[Siorpaes and Hepp, 2008] Siorpaes, K. and Hepp, M. (2008). Games with a purpose for the semantic web. *IEEE Intelligent Systems*, 23(3):50–60.

[Thurlow, 2010] Thurlow, I. (2010). Demonstrator and validation report. Deliverable, ACTIVE.

[Thurlow and Gu, 2011] Thurlow, I. and Gu, J.-Y. (2011). Fully functional prototype and validation report. Deliverable, ACTIVE.

[Tilly, 2010] Tilly, M. (2010). Dynamic models for knowledge processes – final models. Deliverable, ACTIVE.

[Tilly et al., 2010] Tilly, M., Ruiz, C., and Dengler, F. (2010). Context-aware knowledge processes fully functional prototype. Deliverable, ACTIVE.

[Vrandecic et al., 2010] Vrandecic, D., Ratnakar, V., Krötzsch, M., and Gil, Y. (2010). Shortipedia - Aggregating and Curating Semantic Web Data. In Bizer, C. D. M., editor, *Semantic Web Challenge*.

[Warren et al., 2009] Warren, P., Kings, N., Thurlow, I., Davies, J., Bürger, T., Simperl, E., Moreno, C. R., Gomez-Perez, J., Ermolayev, V., Ghani, R., Tilly, M., Bösser, T., and Imtiaz, A. (2009). Improving knowledge worker productivity - the ACTIVE integrated approach. *BT Technology Journal*, 26(2).

A Appendix

A.1 ACTIVE SMW Extensions

Extensions that were developed by KIT:

Name	Contact person	Description	Website/Documentation
Semantic Project Management	Frank Dengler	Project Management Formats for Semantic MediaWiki inline queries (e.g., gantt charts)	http://www.mediawiki.org/wiki/Extension:Semantic_Project_Management
Process Visualization for Semantic Result Formats	Frank Dengler	Included in Semantic Result Formats.	http://www.mediawiki.org/wiki/Extension:Semantic_Result_Formats
Process Editor	Frank Dengler	Oryx-based process editing.	Not released, yet.
ExternalQueries	Basil Ell	Used in CSV-Import extension.	Will be made public until the final review.
SMWWriter	Denny Vrandecic	Used in CSV-Import extension.	http://semantic-mediawiki.org/wiki/Help:SMWWriter
AskTheWiki	Daniel Herzig	This extension allows you to perfom semantic search on your wiki.	AskTheWiki showed limitations, which is why a light-weight version AskQ was developed and will be made available to the public.
AskQ	Daniel Herzig	Provides a semantic keyword query intrepreter and basic faceted browsing capabilities.	http://www.mediawiki.org/wiki/Extension:AskQ
RDFIO	Denny Vrandecic	Used for importing the Enterprise Ontology into SMW.	http://www.mediawiki.org/wiki/Extension:RDFIO

Extensions that were developed by people at KIT but on a case study specific basis. For such extensions, the case study partners decide whether they would like to publish them:

Name	Contact person	Description
ACTIVE extensions	Denny Vrandecic	Connection between ACTIVE server and SMW, including specific functionalities, e.g., active workspace tagging and context ask.
Cadence Discussion extension	Basil Ell	More a number of templates but still possible to publish in the form of SMW usage recommendations.
PortalWSImport	Basil Ell	Import Content from Portal Search WebService.
PropertyBox	Basil Ell	For Testing in MediaWiki environment.
AutomaticAnnotations	Basil Ell	Annotate some known words/phrases from a taxonomy. Each word in an article that is part of the taxonomy is transformed into a tag. Therefore an article becomes tagged with each word
BT-CSV-Import	Basil Ell, Benedikt Kämpgen	Various CSV files – e.g., containing data about proposals, customers or team members – are imported into the wiki and partly displayed non-editable using External Queries extension.

Extensions that were developed by the community, recommended by people at KIT and used within the case studies. These extensions are available for the public and downloadable from Semantic MediaWiki website, MediaWiki website, Vulcan website, or Ontoprise website:

Name	Description
SAMLAuth (Version 0.0.1)	SAMLAuth uses the SimpleSAMLphp libraries and services to provide SSO based authentication.
Semantic Forms (Version 1.8.4)	Forms for adding and editing semantic data
Special:NukeDPL (Version 1.2.1, 2009-03-20)	Mass delete by DPL query
DynamicPageList (Version 1.8.9)	A highly flexible report generator for MediaWikis - manual and examples, see website.
ImageMap	Allows client-side clickable image maps using <imagemap> tag

MagicNoCache (Version 1.1)	Adds a NOCACHE magic word to disable caching of certain pages.
ParserFunctions (Version 1.1.1)	Enhance parser with logical functions
Rich Media Extension (Version 1.3-for-SMW-1.4.x)	The Rich Media Extension provides an ontology to allow easy handling of media such as documents, images, doc, pdf etc. The ontology comprises templates and forms and examples. It enhances a one-click media upload of files and enables annotation of media in a simple way.
Semantic Internal Objects (Version 0.4)	Setting of internal objects in Semantic MediaWiki
Semantic Treeview (Version 1.2)	Improved version of the Mediawiki extension Treeview. Extends the wiki parser to allow bullet and numbered lists to work with recursion and optionally allows these to be rendered as collapsible trees using the free dTree JavaScript tree menu.
Semantic UltraPedia Extension (Version 0.5)	Utilities for UltraPedia.
Semantic WikiTag Extension (Version 2.0)	Webservice support for MS Office WikiTag product.
SMWHalo Extension (Version 1.4.6-for-SMW-1.4.3)	Facilitate the use of Semantic Mediawiki for a large community of non-tech-savvy users. View feature description.
StringFunctions (Version 2.0.3)	Enhances the parser with string functions
URL Get Parameters parser extension (Version 1.0.0)	Provides the {{#urlget:...}} parserfunction which enables access to the url get parameters.
Variables	Define page-scoped variables
Collaboration (Version 1.0)	Some fancy collaboration tools.
Semantic Forms Input Types (Version 0.2)	Additional input types for Semantic Forms.
Semantic Result Formats (Version 1.4.6)	Additional formats for Semantic MediaWiki inline queries
SMW User Manual v1.0	A context sensitive help for SemanticMediaWiki, and other semantic extensions. View online documentation in the SMW+ User Forum.
WYSIWYG extension (Version 1.2, FCK 2.6.4 Build 21629)	FCKeditor for Semantic MediaWiki
Delete Batch (Version 1.1)	Deletes a batch of pages
Graphviz (Version 0.4)	Graphviz (http://www.graphviz.org) is a program/language that allows the creation of numerous types of graphs. This extension allows the embedding of graphviz markup in MediaWiki pages and generates inline
Header Tabs (Version 0.6.5)	Adds tabs to the page separating top-level sections. Originally developed for Ardorado.com
UsabilityInitiative (Version 0.1.1)	Features developed by the Wikipedia Usability Initiative to enhance the usability of MediaWiki
WikiEditor (Version 0.2.0)	Provides an extendable wikitext editing interface and many feature-providing modules
Page Object Model	Adds a set of classes for abstraction of MediaWiki syntax to allow easy extraction and manipulation of pages within other programs

A.2 Enterprise Wikis: Technical Challenges and Opportunities

Enterprise Wikis: Technical Challenges and Opportunities

Benedikt Kämpgen, Basil Ell, Elena Simperl, Denny Vrandečić, Frank Dengler
firstname.lastname@kit.edu

Abstract: Social software has proven valuable in enterprises for collaborative knowledge management. In order to introduce a wiki in the enterprise, we propose a solution that combines Web 2.0 and Semantic Web technologies. We describe how this solution resolves the technical challenges, beyond that, opens up new opportunities, and, also, how it can be realized in a concrete enterprise scenario.

1 Introduction

Social software as a tool for knowledge sharing and collaboration is gaining more and more relevance in the enterprise world [DRBM09]. This especially is true for so-called enterprise wikis, that, just as wikis in the public web, provide their advantages of low usage-barriers and direct benefits within a company intranet. However, simple provision of a Wikipedia-alike does not guarantee acceptance by employees; such wiki software needs to be customized to the specificities of the corporate context. There are technical, social, and organizational challenges to this customization. Human behavior and organizational habits impeding the adoption of social software have been discussed much recently [HDW10]. For instance, a collaboration-unfriendly corporate culture and an unclear value proposition for stakeholders hinder the adoption of Web 2.0 technologies.

Much less is known about the technical challenges related to this adoption. Integration into the enterprise IT landscape and compliance with diverse internally and externally defined policies and regulations are the most obvious examples. Existing literature deals with technical challenges on a high-level without providing useful guidance for enterprises. Also, there are profit-driven systems exclusively fitted to the requirements of enterprises, however, how they solve technical challenges is not known to the public domain. Therefore, we want to elaborate on technical challenges of introducing an enterprise wiki. To do so we build upon our experiences earned in introducing wikis at three globally operating companies in the sectors of telecommunication, consultancy, and electronics design, namely British Telecommunications plc, Accenture, and Cadence Design Systems. These experiences align along the theme of proposal development in response to customer demands, a scenario which is not only highly critical across many business sectors, but also can be seen as a representative example in which *enterprise knowledge structures* are collaboratively created, enriched, and exploited. Together with potential users, we have identified requirements that impose technical challenges to a wiki-based solution.

Ankolekar et al. already have argued that Web 2.0 and the Semantic Web provide complementary technologies [AKTV07]. Therefore, we assume that applying semantic concepts

to an enterprise wiki will not only help to overcome the technical challenges, but also provide new opportunities. More concretely, we propose to use as basis Semantic MediaWiki [KVV+07]. Its usage of standard semantic technologies such as RDF[1] and ontologies provides advantages beginning from an integrated means to formally describe the meaning and organization of the content to various enhancements of the way information is retrieved, displayed and navigated within the wiki. For evaluation, we implemented the solution; the results were again presented to potential users, who have confirmed that the requirements were met.

With this paper, it is our aim to achieve greater awareness of the technically motivated challenges behind enterprise wiki adoption and to allow enterprises to make an informed decision about deployment of our solution in a similar scenario. Developed around representative enterprise knowledge structures – information sources for proposal development –, we assume the technical challenges to be typical for enterprise wikis. The description of the implementation is not only meant for evaluation but also provides a concrete example.

The remainder of this paper is structured as follows. We first introduce the enterprise scenario and describe its relevant requirements (Section 2). In Section 3, we describe how semantic technologies can be used to fulfil the technical challenges. In Section 4 we foster our claim, explaining a concrete implementation. After that, an overview of related work is given in Section 5, followed by conclusions in Section 6.

2 Enterprise scenario

Proposal development in enterprises is commonly perceived as a knowledge-intensive, collaborative process, in which a proposal manager and a team create a description of the products and services delivered by the company at an estimated cost to a potential customer. A proposal includes various types of information – for instance, about marketing, pricing and certification – provided by various enterprise departments – for instance, technical consultants, product specialists and sales persons. The development typically includes activities such as *selecting the proposal team*, *gathering information about the customer*, *discussing customer issues and possible solutions*, and *getting approval for pricing*; still, it is highly variable and its full particulars can hardly be recorded through productivity software which is often used in this context, such as Microsoft Word, Excel and SharePoint, as well as messaging services [STW+ar]. To illustrate this, consider the activity of gathering information about a customer. The way this activity is carried out depends on the preferences and expertise of the proposal development team, and on undocumented social communication and collaboration practices. One might visit a website, consult the intranet portal or call a former colleague, to name just a few.

For engineering of functional and non-functional requirements, about 50 potential users – knowledge workers involved in proposal development – were asked within the ACTIVE project[2] for their opinion. We only include requirements that impose technical challenges

[1] http://www.w3.org/RDF/
[2] http://www.active-project.eu/

to our solution. Similarly, we do not include requirements that by themselves can be solved by typical wiki solutions, e.g., click-to-edit functionality, storage of both text and media, and change tracking.

Functional requirements A proposal development workspace should help users to record, share and collaboratively refine on relevant *enterprise knowledge structures*. Those structures comprise various information sources, available to an enterprise, that its employees use to manage the knowledge for their daily work. Examples include customer descriptions, product specifications, and price lists.

The system should provide guidance to the user about what information is to be put into the wiki. Also, users prefer to not depend on wiki syntax, but to have simpler and faster ways of adding information into the wiki, e.g., forms.

Users should be able to access the results of common activities executed for previous proposals and find such reusable pieces of content in due time.

The wiki is supposed to not only provide the necessary structure, but also to offer concrete information from already existing data sources relevant for proposal development. Enterprises contain many different data sources, for instance relational databases, content management systems, and various document formats. They contain more or less unstructured information. Examples include descriptions of finished proposals; price lists of competitors; reports about industry sectors; and other elements that employees use for proposal development and therefore want to access, discuss, and refine through the wiki.

Data quality is of high importance. Users will not adopt the wiki if incorrect information is contained and not distinguishable from relevant information. On the one hand, flexibility of what to put into the wiki should be preserved. On the other hand, it should be possible to discover and solve data problems.

Employees are accustomed to tools that help with developing proposals. For instance, the end-proposal is usually delivered in form of a Microsoft Word document. In the ideal case, users are free to continue using their tools, but these tools are extended and allow to exploit the wiki's added values. Also, as interaction in a wiki occurs asynchronously, some users prefer to be able to keep track of changes without constantly visiting the wiki.

Non-functional requirements The acceptance is likely to be higher if the tool is intuitive to use, also by users without technical background, and minimally invasive to established workflows and the enterprise IT landscape. The system should run sufficiently fast, with loading times similar to external webpages.

3 Solution based on a semantic wiki

In this section, we argue that semantic technologies will help to overcome much of the technical challenges imposed by our proposal development scenario, and beyond that, open up new opportunities. As a reference semantic wiki, we have chosen to use Semantic

MediaWiki (SMW). Then, in the next section, we present an implementation of a proposal development wiki having these capabilities.

Creating structured information Our proposal development scenario makes it necessary to capture and refine enterprise knowledge structures within the wiki. For that, guidance to the users is necessary. Information stored in SMW conforms to machine-readable RDF. More understandable, it allows to have property-value pairs explicitly assigned to wiki pages. Such a property-value pair can be a named link (so-called *object property*) to another page, e.g. "locatedInCountry" Page of country, but can also be a typed attribute (so-called *datatype property*), e.g., "hasTag" String, "hasFoundingDate" Date, and "hasHeight" Number. *Properties* can be inserted into a page with wiki syntax, but also using forms. First, this makes them easily usable. Second, enterprise knowledge structures, as we have them in our proposal development scenario, can be defined through *categories* of pages (so-called *classes*) with certain properties and serve as guidance for the users of how to use the wiki and what structured information to capture in it, e.g., proposal with a team, a customer, and a due date.

The structure can first be modelled in standard knowledge representation languages for the Semantic Web, such as RDF, and OWL[3]. Depending on the expressivity, this *ontology* can be automatically or manually applied to the wiki [VK06]. Note, such enterprise knowledge structures are bound to continuous change and refinement, e.g., due to changes to enterprise workflows. Semantic data structures, in contrast to relational data structures, can be extended at any time in SMW either by administrators or the users themselves without modifying previous contributions.

Retrieving information Users need to retrieve specific information from these enterprise knowledge structures. The machine-readable information stored with SMW gives more sophisticated possibilities of retrieving data from the wiki, other than traditional keyword searches. First, it can be specifically asked for certain properties of a page, e.g. the customer of a proposal. Second, all pages of a certain category having certain properties can be listed as an overview, including links to those pages, e.g, all products within a specific price range. Various result formats can be used, starting from simple tables to more advanced calendars, timelines, and maps. Also, *facetted search* is possible, for incrementally filtering lists of pages with keywords or property-ranges (see Figure 1 in a later section). Third, more complex but still user-friendly querying similar to the standard semantic web querying language SPARQL[4] is possible. The users can enter keywords, the system looks for connections between pages described with the keywords and lists those pages [HHMT09]. In SMW, these queries are possible through forms on special pages, but can also be embedded as *inline queries* in normal wiki pages.

Integrating external information Integrating external source allows to merge its content with existing enterprise knowledge structures, i.e. adding it as pages or properties,

[3] http://www.w3.org/TR/owl2-overview/
[4] http://www.w3.org/TR/rdf-sparql-query/

referencing it from other pages, and visualizing it in new ways. However, simply creating a page for each element within an external source and copying its data from there, will lead to difficulties searching and using this data. SMW provides a possibilty to tightly intergrate external sources. Enterprise knowledge seldomly is represented in RDF, but there are many tools[5] available to transform the formalized enterprise knowledge into RDF, which then can be mapped to specific knowledge structure elements in the wiki, e.g. categories, single pages, and properties. The more structured this external information is, the more this transformation can be done automatically. To deal with redundancy, it is possible to allow users to refer to, and comment upon external sources in the wiki, while changes may be undertaken only through the original systems and tools.

Not only data sources within an enterprise but also sources in the web such as Freebase or other Semantic MediaWiki installations can be integrated, this way. A growing number of web services offer data in RDF[6]. Using explicit bindings to such externally stored sources, wiki pages could be easily enriched with their data. The users may integrate new external sources by themselves, although this can be restricted, e.g, for security reasons.

Improving data quality One of the most useful features of a semantic wiki is its ability to perform consistency checks on the enterprise knowledge structures represented within the wiki and to indicate data quality problems. This provides a means to identify missing or incorrect information, which applies to both genuine wiki content and content from external sources. Users may not directly correct the latter, but they can rate it, and comment on it for revision. Besides the possibility that users detect inconsistencies, some checks can be performed automatically. Deduction methods on the enterprise knowledge structures can provide insights about the wrong usage of categories, pages, and properties [Vra09]. Most of such errors cannot be automatically repaired, but at least, made visible to the users or administrators. For example if the imported data contains information about a proposal with customer X and a wiki page exists about X, which is not a member of the customer category, adding that page to the category can be automatically suggested to the administrator.

Interplay with other enterprise tools Also, enterprise knowledge structures reach full potential if they are not stored in an isolated data silo but can be accessed from other enterprise tools. SMW allows not only to integrate external sources in a standard and machine-readable manner, but generally to use external tools to input from and output to the wiki. The content of a semantic wiki can be extracted as RDF, as well as many other structured data formats, e.g., JSON, vCard, and BibTeX. Results of queries can be regularly checked for new pages or for modified properties and published as RSS-feeds or send per e-mail. Using HTTP requests to the wiki, external tools such as Microsoft Word can access, add, or modify pages and properties.

Semantic technologies cannot generally fulfil the non-functional requirements of our scenario. Those are implementation-dependent and therefore will be described in the next

[5] `http://www.w3.org/RDF/`
[6] `http://www.linkeddata.org/`

section, where we evaluate our solution.

4 Implementation of a proposal development workspace

As it is our aim to have general guidelines for applying our solution, we do not depict differences between the wikis we realized for our three case studies, but focus on the lessons learned from the three implementations and describe and explain it as one. Our implementation is based on the open-source wiki software MediaWiki and its semantic counterpart Semantic MediaWiki, which have been augmented with a series of general-purpose extensions developed by the community[7] and custom extensions tailored to the needs of the proposal development scenario.

Users access the wiki from the intranet using a personalized and enterprise-wide login, realized through *LDAP Authentication* extension. Pages are cached until they are changed or reprocessing is explicitly requested. This way, SMW leads to not much loss in performance in comparison to a pure MediaWiki installation [HE10].

Creating proposal development information As a workspace for proposal development the wiki supports the entire life cycle of a proposal. For that, we have developed an ontology describing the proposal development structures. It contains categories of pages (classes) such as proposal, team, person, customer, customer issue, discussion question or answer, product, and event. Each of these categories are further defined through properties. For instance, a proposal has one or more proposal sections, a team of persons, a customer with customer issues, a monetary value, and offers one or more products; a customer issue can be discussed through questions and answers; a customer is related to an industry sector; and several meetings are held for a proposal. The ontology has been developed together with potential users, but also by reuse of available enterprise vocabularies and ontologies such as "The Enterprise Ontology" [UKMZ98]. We expect that continuous refinement of the ontology is necessary, also after an official launch of the system, and will mainly be done by administrators.

Each relevant element of a proposal is represented in the wiki as a dedicated page or part of a page. Users can easily create pages and property-value pairs through the *Semantic Forms* extension. Adding properties to proposal elements supports auto-completion, checkboxes, radio buttons, and other value selection widgets such as mini-calendars and map views. Furthermore, the extension *Header Tabs* facilitate the realization of forms that are similar in their appearance to the rendered pages. Discussions, e.g, about customer issues, have a structure similar to forum applications.

Retrieving proposal information In order to provide users with an overview of the proposal workspace, pages explaining and listing a particular aspect of proposal development

[7] openly available from `http://www.mediawiki.org/wiki/Extension_Matrix` (MediaWiki) and `http://semantic-mediawiki.org/wiki/Help:Extensions` (SMW)

are linked from the wiki main page. Administrators define queries and visualizations, e.g. listing all open proposals, all high-priority customer issues, and all products of a certain category. The users themselves can issue queries and store them on wiki pages, although we expect more use of keyword based searches that, with *AskTheWiki* [HHMT09] extension, still exploit the wiki's structure. Figure 1 shows the workspace customized through *Halo* extension, and an anonymized example of facetted search using *Exhibit*.

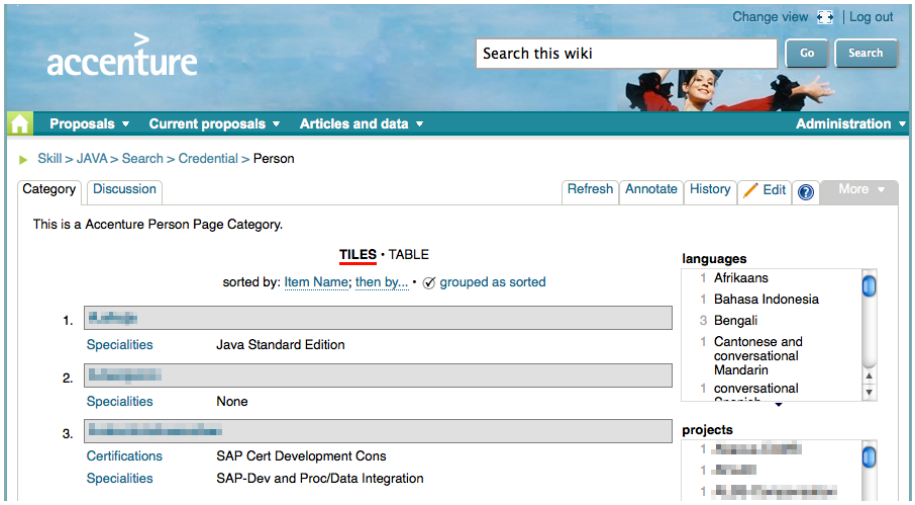

Figure 1: Anonymized wiki content displayed through facetted search.

Unnecessary searching and browsing is reduced by sharing relevant information between wiki pages. For instance, the property "hasCustomerVision" is not only shown on the particular customer page but also on proposal pages concerning this customer.

Integrating proposal-related information The proposal development structures established in the wiki were partly populated by content from external sources. We integrated externally created information about customers, industry sectors, offered products, time and pricing, people and their involvement in proposals, as well as status and result of proposals. This data mainly comes from relational databases or excel sheets, and is preprocessed, e.g., different date formats are matched.

Some of this information is first transformed in an RDF-compliant format, then imported as pages such as proposals, or properties such as a customer vision. Other imported wiki content such as person identification number is not supposed to be redundantly stored and open to changes. This information is only visualized, allowing users to refer to, and comment upon it in the wiki, while changes may be undertaken only through the original source. Although promising for the future, so far, we have not found useful Linked Open Data for proposal development.

Improving proposal data quality The system does not only allow input and retrieval of new or existing information, it also ensures long-term data quality. Integrating various external data sources into the wiki makes implicit connections between them explicit. For instance, proposal descriptions may contain team member names, that provide direct links to pages of those persons. Misspelled names can be easily identified through links to non-existing pages. Correct abbreviations or accronyms can be included as synonyms. More data quality problems are identified and corrected by the administrators when they export the data as RDF using *RDFIO* extension and compare its consistency with the proposal development ontology. That way, wrong properties can be detected and made visible to the users, e.g., an event having a monetary value; a customer being located in a proposal; and a due date containing a number.

Interplay with tools for proposal development For automatic notifications about certain changes in the wiki, users can subscribe to RSS-feeds. Those publish information about new pages, e.g., such as new customer issues with a high-priority, or more fine-grained information about modified properties such as the monetary value of a proposal. Such information can also be displayed as a widget on other company intranet sites, and, using *Semantic Notifications* extension sent per e-mail.

When finally creating the actual proposal document within Microsoft Word, the users can access the structured information in the wiki through the *WikiTags* extension. If a particular proposal element is mentioned in the document, Word automatically underlines it and provides a wiki-based context menu for it, e.g., for fast copy-paste-like insertion of properties stored in the wiki (Figure 2). Vice versa, for new expressions not yet used in the wiki, it is possible to create wiki pages, properties and links directly from the office application. Once additional information about these is collaboratively assembled in the wiki, the results can be used in the Word document.

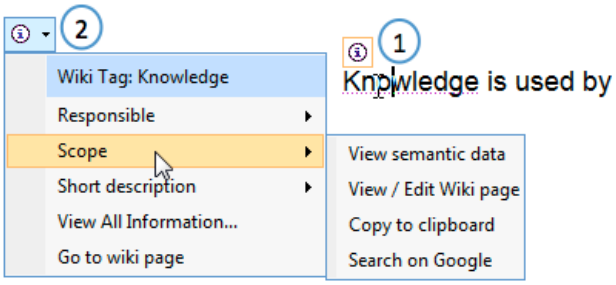

Figure 2: Microsoft Word underlining a word (1) and providing a wiki-based context menu (2).

Within the ACTIVE project, the implemented solution was presented to the potential users, who have confirmed that it fulfils the requirements. Recently, trials have been initiated for evaluating the solution with actual users.

5 Related work

Not much public information can be found on technical challenges of enterprise wikis. Danis and Singer [DS08] present a detailed study of the differences in wiki utilization in corporate, educational and public settings, but without going into detail on technical aspects. If any, only general solutions to technical challenges can be found. For instance, deciding what hardware or service to use, and offering features such as a more simple file upload and an indexing of attachments for built-in search [Arc10].

The social software market offers various enterprise systems [DRBM09] that among other things provide wiki solutions. IBM Lotus Connections and Microsoft's SharePoint Server, as market leaders, still lack strong wiki functionality. Atlassian Confluence and TWiki may provide this, however, they do not give details about their approaches to technical challenges. KiWi is a wiki solution that combines Web 2.0 and Semantic Web technologies similarly to SMW and can be used to build enterprise applications [SMSK10].

6 Conclusions

In this work, we have proposed and implemented a semantic-wiki-based solution for introducing a wiki in a typical enterprise scenario. It fulfils the technical requirements and beyond that opens up new opportunities for the enterprise, e.g., exploitation of the growing Linked Open Data and seamless interplay between the wiki and external tools.

We now conclude with some final remarks and possible future work. Due to the large number of extensions, choosing the most appropriate ones is a tedious task, that could be improved by further empirical studies on the utility of existing or possible extensions in specific application scenarios. For now, we have evaluated the hypothesis that the technical challenges can be resolved by a semantic wiki. We have not evaluated whether the requirements acually were representative and whether our solution will bring the expected benefits. Measuring its impact is under work in recently initiated trials. As an example, the time the user spend with the wiki, either directly or through external tools, could be an indicator for sucess. Still, evaluating the success of a wiki is not an easy task, due to many confounding success factors. We have not tried to fulfil the requirements with other technologies than Semantic MediaWiki and, thus, cannot say much about comparison to other systems, be they semantic wikis or not. As intended, we solely argue that semantic wiki technologies clearly expand the capabilities of knowledge workers for managing enterprise knowledge structures.

Acknowledgements

Work presented in this paper has been funded by the EU IST FP7 project ACTIVE under the grant 215040.

References

[AKTV07] Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, and Denny Vrandečić. The two cultures: mashing up web 2.0 and the semantic web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 825–834, New York, NY, USA, 2007. ACM Press.

[Arc10] Bill Arconati. Factors for the success for Wikis 2: Organization is the key (blog post), 2010. `http://blogs.atlassian.com/confluence/2010/06/factors-for-success-for-wikis-2-organization-is-the-key.html`, Last visit 07/11/2010.

[DRBM09] Nikos Drakos, Carol Rozwell, Anthony Bradley, and Jeffrey Mann. Magic Quadrant for Social Software in the Workplace. Gartner RAS Core Research Note G00171792, Gartner, 10 2009.

[DS08] Catalina Danis and David Singer. A wiki instance in the enterprise: opportunities, concerns and reality. In *CSCW '08: Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 495–504, New York, NY, USA, 2008. ACM.

[HDW10] Lester J. Holtzblatt, Laurie E. Damianos, and Daniel Weiss. Factors impeding Wiki use in the enterprise: a case study. In *CHI EA '10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 4661–4676, New York, NY, USA, 2010. ACM.

[HE10] Daniel Herzig and Basil Ell. Semantic MediaWiki in Operation: Experiences with Building a Semantic Portal. In *9th International Semantic Web Conference (ISWC2010)*, November 2010.

[HHMT09] Peter Haase, Daniel M. Herzig, Mark Musen, and Duc Thanh Tran. Semantic Wiki Search. In *6th Annual European Semantic Web Conference, ESWC2009, Heraklion, Crete, Greece*, volume 5554 of *LNCS*, pages 445–460. Springer Verlag, Juni 2009.

[KVV+07] Markus Krötzsch, Denny Vrandečić, Max Völkel, Heiko Haller, and Rudi Studer. Semantic Wikipedia. *Journal of Web Semantics*, 5:251–261, September 2007.

[SMSK10] Rolf Sint, Mark Markus, Sebastian Schaffert, and Thomas Kurz. Ideator - a collaborative enterprise idea management tool powered by KiWi. In *5th Workshop on Semantic Wikis: Linking Data and People (SemWiki2010)*, 2010.

[STW+ar] Elena Simperl, Ian Thurlow, Paul Warren, Frank Dengler, John Davies, Dunja Mladenic, Jose Manuel Gomez-Perez, Carlos Ruiz Moreno, and Marko Grobelnik. Overcoming Information Overload in the Enterprise: the ACTIVE Approach. *IEEE Internet Computing*, November-December 2010 (to appear).

[UKMZ98] Mike Uschold, Martin King, Stuart Moralee, and Yannis Zorgios. The Enterprise Ontology. *Knowl. Eng. Rev.*, 13:31–89, March 1998.

[VK06] Denny Vrandečić and Markus Krötzsch. Reusing Ontological Background Knowledge in Semantic Wikis. In Max Völkel and Sebastian Schaffert, editors, *Proceedings of the First Workshop on Semantic Wikis – From Wiki To Semantics*, Workshop on Semantic Wikis. AIFB, ESWC2006, June 2006.

[Vra09] Denny Vrandečić. Towards Automatic Content Quality Checks in Semantic Wikis. In *Social Semantic Web: Where Web 2.0 Meets Web 3.0*, AAAI Spring Symposium 2009, Stanford, CA, March 2009. Springer.

[end of document]