CSW
Corporate Semantic Web

Corporate Semantic Web
Report V

Prototypical Implementations
Working Packages in Project Phase II

Technical Report TR-B-12-04

Adrian Paschke, Gökhan Coskun, Ralf Heese, Radoslaw Oldakowski,
Mario Rothe, Ralph Schäfermeier, Olga Streibel, Kia Teymourian
and Alexandru Todor

Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Semantic Web Group

16 May 2012

Freie Universität Berlin

INNOPROFILE
UNTERNEHMEN REGION
Die BMBF-Innovationsinitiative
Neue Länder

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Corporate Semantic Web:
Report V

Adrian Paschke, Gökhan Coskun, Ralf Heese,
Radoslaw Oldakowski, Mario Rothe, Ralph Schäfermeier
Olga Streibel, Kia Teymourian and Alexandru Todor

Freie Universität Berlin
Department of Mathematics and Computer Science
Corporate Semantic Web Group
Königin-Luise-Str. 24-26
14195 Berlin, Germany

paschke,coskun,heese,oldakowski,mrothe,
schaef,streibel,kia,todor@inf.fu-berlin.de

Technical Report TR-B-12-04

16 May 2012

Abstract

In this technical report, we present the concepts and first prototypical implementations of innovative tools and methods for personalized and contextualized (multimedia) search, collaborative ontology evolution, ontology evaluation and cost models, and dynamic access and trends in distributed (semantic) knowledge. The concepts and prototypes are based on the state of art analysis and identified requirements in the CSW report IV [43].

Contents

Chapter 1

Introduction

Corporate Semantic Web (CSW) deals with the application of Semantic Web technologies (in particular rules and ontologies) within enterprise settings. It address the technological aspects of knowledge engineering and managing semantic enabled IT infrastructure to support (collaborative) workflows, communication, knowledge management, and (business) process management in enterprises. But, it also addresses the pragmatic aspect of actually using Semantic Web technologies in semantic enterprises.

In the first phase of the BMBF funded InnoProfile project *Corporate Semantic Web* parts of the CSW vision have been realized - see [16, 42, 41]. The second phase of the CSW project addresses several working packages which research and develop advanced methods and tools for personalized and multi-media search, distributed collaborative knowledge engineering and evolution, as well as the pragmatic aspect of quantifying the use of ontologies in terms of engineering cost models and qualifying their quality in terms of evaluation methods. (see appendix Work Packages A). In CSW report IV [43] we have described the underlying research problems and analyzed the state of art of existing problem solutions. Based on the derived requirements from the last report, this report will describe concepts and first prototypes of tools addressing these problems. The further report is structured along the three research pillars of the CSW approach - *semantic engineering*, *semantic collaboration*, *semantic search.*

Chapter 2

Corporate Ontology Engineering

2.1 Cost Models for Ontologies (AP11)

In our last technical report analyzing the state of art of estimating the costs for ontology development [43], we introduced *ONTOCOM* (*ONTO*logy *CO*st *M*odel) [44]. While algorithmic cost models such as *ONTOCOM* are easy to apply, they lack in accuracy.

In this working package, we combine *ONTOCOM* with project monitoring metrics emerged from agile development processes in order to achieve an initial, albeit inaccurate, cost estimation for an envisaged project and refine the cost predictions in the course of the project, using actual project runtime data.

In the following sections, we give a detailed problem statement, then we introduce our hybrid, self-adapting cost model and present the architecture of a prototypical implementation.

2.1.1 ONTOCOM

ONTOCOM is an algorithmic cost model derived from the software cost estimation model *COCOMO* [10, 11]. Algorithmic cost models employ a mathematical function, mapping from several known numeric or ordinal input parameters to a cost value, typically expressed in person months. Like most algorithmic cost models, *ONTOCOM* was derived from historical project data and calibrated using different statistical methods, such as multivariate regression, and bayesian or ANOVA analysis.

A first version of *ONTOCOM* was based on empirical data from 36 Ontology Engineering projects. In a second pass, the data set has been extended to 148 projects [51]. The *ONTOCOM* model considers a number of ordinal cost drivers which are supposed to influence the overall cost of an ontology development project and which appear as weighting factors in the cost function. The calibrated results from the second survey suggest that from 11 cost drivers only six explain most of the behavior of the model. These are:

- *Domain Analysis Complexity (DCPLX)*: accounts for those features of the application setting which influence the complexity of the engineering outcomes,

- *Evaluation Complexity (OE)*: accounts for the additional efforts eventually invested in generating test cases and evaluating test results,
- *Ontologist/Domain Expert Capability (OCAP/DECAP)*: accounts for the perceived ability and efficiency of the single actors involved in the process (ontologist and domain expert) as well as their teamwork capabilities,
- *Documentation Needs (DOCU)*: states the additional costs caused by high documentation requirements,
- *Language/Tool Experience (LEXP/TEXP)*: measures the level of experience of the project team w. r. t. the representation language and the ontology management tools, and
- *Personnel Continuity (PCON)*: mirrors the frequency of the personnel changes in the team.

Strenghts and Shortcomings of Algorithmic Cost Models

Algorithmic cost models such as *ONTOCOM* have the advantage of being relatively easy to apply, as long as all required input parameters are known. They rely on historical data from previous projects and are therefore able to yield an initial estimate prior to project kick-off, without requiring deep insight into the nature of the project in question.

However, not all factors with a potential impact on a project's cost might be known at the initial stage. Project complexity can be underestimated or requirements can change at a later stage because they were not obvious in the first place and only became apparent with growing insight into the problem at development time. Other factors which are hard to predict comprise management and personnel related issues such as withdrawal of team members due to priority changes, unexpected termination, or sickness.

Algorithmic cost models are not able to respond to this kind of unpredictable changes, limiting them in their flexibility as well as in their accuracy, with error rates greater than 100 % [30, 15].

In order to overcome these weaknesses, while still benefiting from the above mentioned advantages of arithmetic cost models, we have developed a hybrid model which is applicable to agile development processes. Our hybrid model is based on *ONTOCOM* at the initial stage of a project. The novelty lies in a feedback cycle which gradually refines the initial estimation during the course of the development, using actual project cost data.

2.1.2 Agile Ontology Development

As stated in [43], small and mid-sized companies seek lightweight and dynamic methods and practices for ontology development and maintenance with minimal need for ontology experts involved. Based on this insight, we adopt agile principles and practices from the software engineering domain to the process of ontology development. In the specific part of our work described in this report, we focus on modeling cost in the context of agile ontology development.

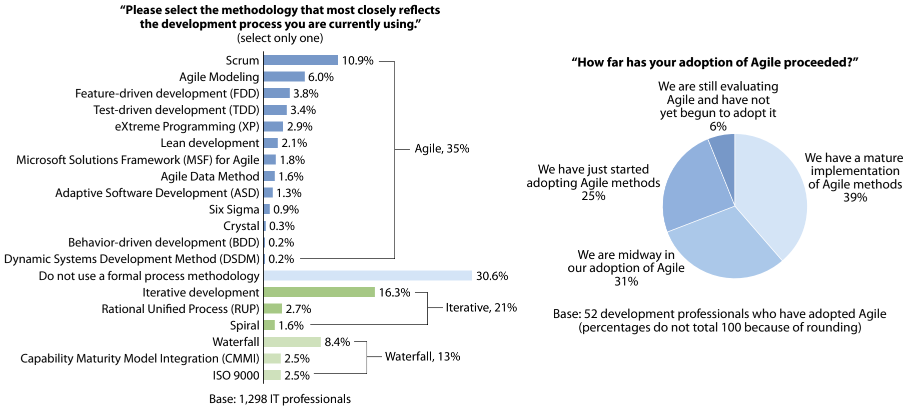
Agile Development

Agile development is defined by a set of values, principles and practices which are supposed to circumvent the administrative overhead caused by methodological rigidness of classical development models, such as the linear waterfall model.

These values are in particular individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [5].

Based on these values, a set of principles has been derived, including rapid and frequent delivery, simplicity, welcoming changing requirements, even late in development, working software as the principal measure of progress, and adaptation to changing circumstances [5].

A survey conducted by Forrester among 1,298 IT professionals in late 2009 [62] shows that over one third of the consulted enterprises have effectively adopted or are in the process of adopting agile development methods (Figure 2.1a). 70 % of these enterprises consider the state of adoption as being midway or mature (Figure 2.1b).



(a) Rate of adoption of agile practices

(b) State of maturity of agile practices

Figure 2.1: State of adoption and maturity of agile software development practices in IT organizations. Source: The Forrester Wave. Develop Management Tools, Q2 2010

For these reasons, and as we have argued throughout our last reports, we believe that agile development principles and practices are the suitable methodological means for small and mid-sized organizations which do not possess the necessary resources to engage into heavy-weight planning and development processes.

Agile Practices and Ontology Development

Among the practices following from the values and principles mentioned in the previous section, in particular those relevant in the context of this work, are rapid release cycles, lean requirements engineering, and test driven development. Auer et al., in their work on the agile ontology development methodology *RapidOWL*, derive further practices especially adapted to the field of agile ontology development, some of which are information integration, ontology evolution, and consistency checking [4].

These practices comprise activities which can be related to the stages of the outer engineering cycle in our COLM lifecycle model (Figure 2.2). The entire outer cycle can be interpreted in the terms of the agile practice rapid release cycles, where a number of iterations in the outer cycle constitutes a release. The actual release is then represented by the transition into the inner usage cycle.

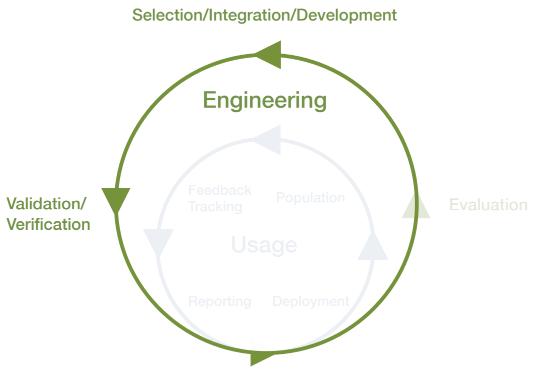


Figure 2.2: The engineering cycle of COLM

The first stage of the engineering cycle consists of the activities selection, integration, and development. In terms of agile practices, this would mean that requirements elicitation, information integration, and the actual development take place at this stage.

The practice test-driven development, which would include the activity of consistency checking but can also be extended to other practices, such as unit testing (cf. [29]), can be assigned to the validation/verification stage in COLM.

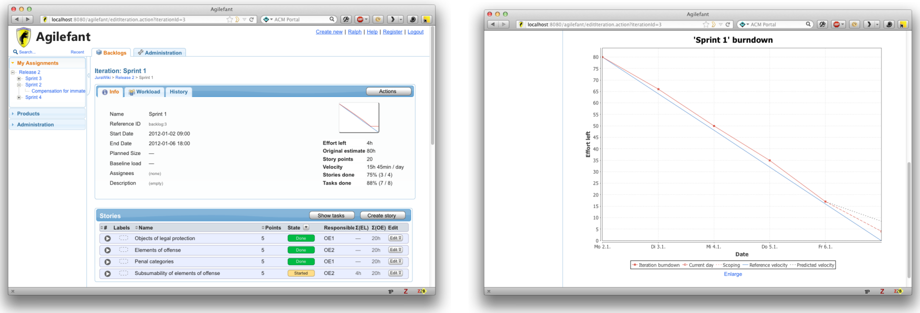
2.1.3 A Self-Calibrating Cost Model

The Forrester study cited in section 2.1.2 identifies *SCRUM* as the agile method the most enterprises have adopted (see Figure 2.1a). Based on this finding, we apply a set of activities defined by the SCRUM methodology to our ontology development lifecycle. For a prototypical implementation, we extended the open source tool *agilefant* [1].

SCRUM uses the notion of user stories for lean requirements elicitation. A user story describes an expected behavior of a part of a system or model from a user's point of view. In the field of ontology development, Suarez-Figueroa et al. propose the utilization of competency questions for the specifications of requirements for an ontology under development [58]. We follow this approach and use user stories in the form of competency questions in the context of our work.

For each story, a number of tasks can be specified and assigned to team members. The effort needed for each story and task is estimated by the team in terms of story points. Story points are dimensionless and used as a simplified

[1] `http://www.agilefant.org/`



Iteration: Sprint 1

JuraWiki > Release 2 > Sprint 1

Name	Sprint 1
Reference ID	backlog:3
Start Date	2012-01-02 09:00
End Date	2012-01-06 18:00
Planned Size	—
Baseline load	—
Assignees	(none)
Description	(empty)

Effort left	4h
Original estimate	80h
Story points	20
Velocity	15h 45min / day
Stories done	75% (3 / 4)
Tasks done	88% (7 / 8)

Stories

#	Labels	Name	Points	State	Responsible	Σ(EL)	Σ(OE)	Edit
		Objects of legal protection	5	Done	OE1	—	20h	Edit
		Elements of offense	5	Done	OE2	—	20h	Edit
		Penal categories	5	Done	OE1	—	20h	Edit
		Subsumability of elements of offense	5	Started	OE2	4h	20h	Edit

(a) User stories and effort related statistics

(b) A burndown chart

Figure 2.3: Project metrics in agilefant

unit for effort estimation. The rationale behind this practice is that estimating effort relative to another task is simpler than estimating absolute effort in terms of hours, days, or months.

As soon as all stories and derived tasks have been collected and estimated, the workload is divided and assigned to subsequent iterations, or sprints. During each sprint, the team members work on their tasks. The goal of each sprint is to get the assigned work done and to produce a functional part of the product. A subsequent number of sprints leads to a feature-complete and tested product ready for release.

During the sprints, the team members regularly update their progress and the effort they have spent in the project management software. On the basis of these entries, the software can produce a burndown chart which gives an overview of the overall progress and a possible divergence from the schedule (see Figure 2.3b). An important measure in this context is the *project velocity* which denotes the completed story points over a period of time (see Figure 2.3a).

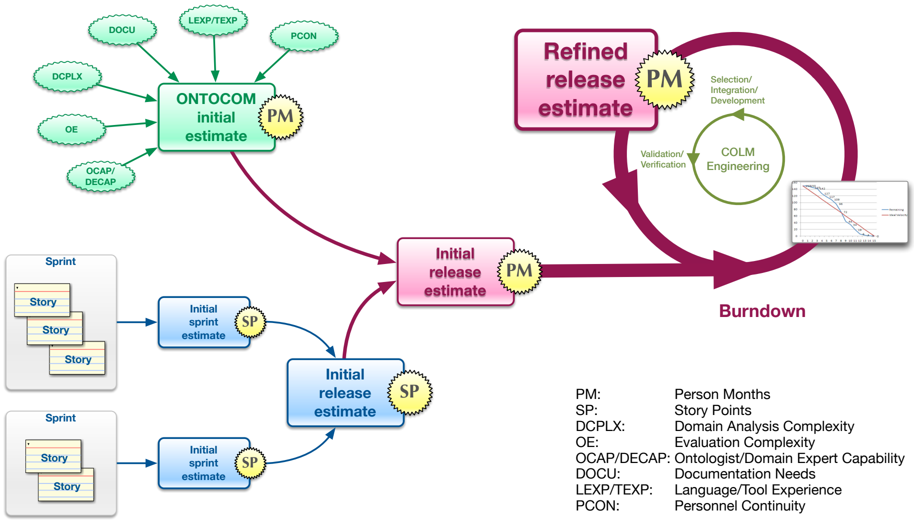
In this work, we used the idea of the burndown chart and the velocity measure in order to calibrate an initial cost estimate achieved by using ONTOCOM. While the initial ONTOCOM estimate lacks reliable accuracy, the estimates by the team members expressed in story points are affected by the problem that there is no mapping between story points and real time units. Our self-calibrating cost model takes the story estimates and normalizes them by using the initial ONTOCOM estimate, yielding a rough estimate for each story in terms of workdays our hours. During each iteration, the prediction is adapted by calculating the current project velocity.

In case of a significant discrepancy between the estimated and the actual project effort, the team leader is asked to assess the possible factors for the discrepancy at the end of the release cycle, where the factors correspond to the cost drivers used by ONTOCOM. This assessment is then transferred back to the ONTOCOM database and used for calibration of the cost factors.

Our model is depicted in figure 2.4.

2.1.4 Prototype

The architecture of our system is depicted in Figure 2.5.



PM:	Person Months
SP:	Story Points
DCPLX:	Domain Analysis Complexity
OE:	Evaluation Complexity
OCAP/DECAP:	Ontologist/Domain Expert Capability
DOCU:	Documentation Needs
LEXP/TEXP:	Language/Tool Experience
PCON:	Personnel Continuity

Figure 2.4: Our proposed hybrid self-adapting cost model

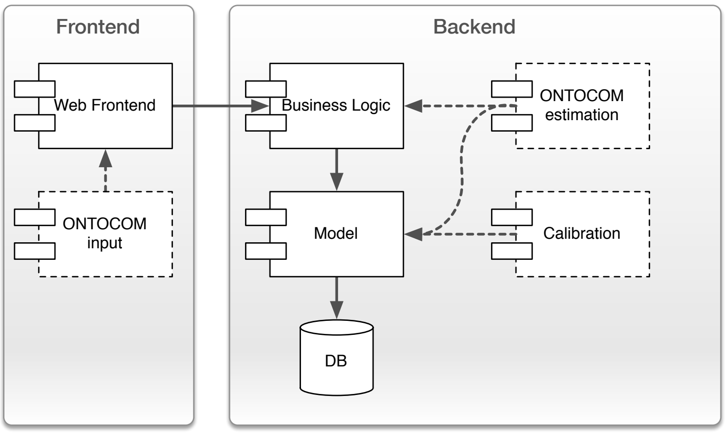


Figure 2.5: The top-level architecture of agilefant and our additional components

As stated in the previous section, we have extended the agilefant agile project planning system and added components responsible for assessing project data relevant for the initial ONTOCOM estimation, for transformation of story point based estimate to work days or work hours, and for the calibration of the initial estimate.

We have integrated the ONTOCOM estimation algorithm and the statistical database into agilefant and extended the user interface in order to allow the input of cost drive related information at the beginning and the end of a project. Further, we have added automated filling of effort estimation values in the story and task entry forms.

2.1.5 Conclusion

In this section we described the architecture and a prototypical implementation of an agile project management tool using a hybrid self-adjusting cost model for project effort estimation in the field of ontology development. The cost model is based on ONTOCOM and dynamic project monitoring metrics used in agile scenarios.

In a next step we will evaluate the tool and its underlying concepts with one of our industrial partners.

2.2 Ontology Evaluation (AP 12)

In previous work [43], we defined two evaluation aspects of the Corporate Ontology Lifecycle Methodology (COLM) [16], namely: (1) semantic evaluation as a part of the ontology selection and modularization process for creating the first version (which is mentioned implicitly in the *Selection/Integration/Development* phase of COLM) and (2) context evaluation as an important part of the continuously ontology maintenance and optimization process (which is mentioned explicitly in COLM). We call the first aspect *Evaluation for Reuse* and the second aspect *Evaluation for Refinement.* In both cases a human-driven approach seems to be suitable at most. We therefore concentrate on appropriate visualization techniques to support the user during the evaluation process.

As proposed in [43] a structure-based approach for both aspects were selected. Thus, it is important to understand what the structure of an ontology is (which is discussed in the next section), before representing the work on *Evaluation for Reuse* in section 2.2.2 and on *Evaluation for Refinement* in section 2.2.3.

2.2.1 Representing Ontologies as Graphs

In the Semantic Web, ontologies are mostly represented by the Web Ontology Language (OWL) based upon the Resource Description Framework (RDF)[2]. RDF allows representing information as triples following the form (Subject, Predicate, Object). The graph syntax of RDF[3] maps triples to graphs where the subjects and the objects are nodes and the predicates are directed edges (from subject to object). At this level the inherent semantics of OWL ontologies

[2] http://www.w3.org/TR/owl-semantics/mapping.html describes how OWL is mapped to RDF

[3] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

are not taken into consideration. The nodes and edges have different types, which are reflected in their labels (namespace and localname). Since the subjects as well as the predicates of RDF statements need to be resources and objects might also be resources, it is impossible to organize the edges and nodes into disjoint sets. This is because a resource, which is a subject or an object in one statement might be used as a predicate in another statement. This problem of the RDF graph representation of triples can be avoided if every named entity of the ontology is represented as a node (even the predicate becomes a node, which is connected with the subject and the object). However, since typically the number of properties is much less than the number of resources which are used as subjects and objects, this graph representation would lead to a graph structure in which the properties are central nodes with high degree values. Some predicates such as "hasLabel" or "hasComment" would have a high centrality value. Hence, it is important to filter and remove such concepts, which have a significant impact on the graph structure analysis of an ontology, but which are not necessary in order to understand the content of an ontology. Furthermore, it is important to take different namespaces into consideration.

We developed three basic approaches how to represent an ontology as a graph. Firstly, the RDF graph syntax is used as it is, that means the subject and object of each statement are nodes, while the predicate is the connecting edge, directed from the subject to the object (variant V1).

Secondly, the predicates are also represented by nodes, where two unlabeled directed edges are created. One edge is directed from the subject to the predicate, while the second is directed from the predicate to the object (V2).

Thirdly, a graph is created where only classes are represented as nodes connected by properties as edges, where the direction is from the domain class of the property to the range class of the property (V3). This variant is based on the idea, that classes are the major objects of an ontology, while the properties can be seen as extensions of those classes and depend on them. Figure 2.6 shows the FOAF ontology represented according to the three mentioned graph variants.

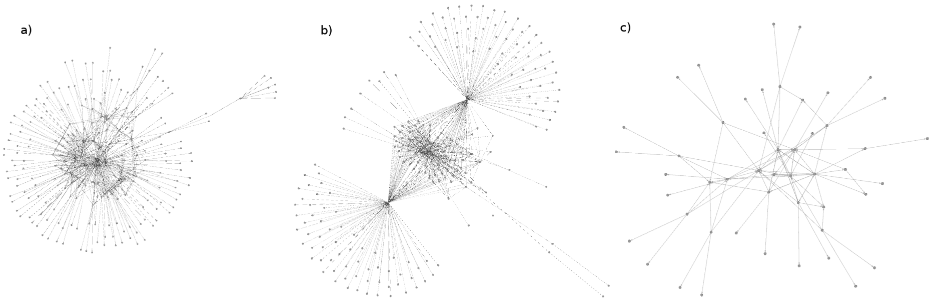


Figure 2.6: FOAF represented in a) graph variant V1, b) graph variant V2, and c) graph variant V3

There are also two different extensions of these variants. In the first extension (named as VxL) the literals are filtered during the graph creation process. This filter is enhanced by the second extension (named as VxLX) by excluding concepts with external namespaces. Summing up, for our experiments we created nine different graph variants for each ontology. The different variants are shown in

Table 2.1.

Table 2.1: Different graph representations for ontologies

Variant name	Description
V1	Plain RDF graph
V1L	as V1 without literals
V1LX	as V1L without external namespaces
V2	Plain RDF graph, but predicates are represented as nodes
V2L	as V2 without literals
V2LX	as V2L without external namespaces
V3	Class graph
V3L	as V3 without literals
V3LX	as V3L without external namespaces

2.2.2 Evaluation for Reuse

Ontology reuse attracts the interest of the Semantic Web community and its current pragmatic version the Linked Data community where ontologies are considered as shared knowledge and are interlinked. Even though ontology reuse is part of various ontology engineering methodologies there are no best practice solutions which describe how existing ontologies should be analyzed and evaluated for their (re)usability. In the field of software engineering component-based development and appropriate documentations are established methods to support reuse. While the adoption of software engineering methods to ontology engineering has been addressed in various scientific publications within the ontology engineering community (e.g. [23], [21]) the documentation-based reuse has not been tackled in depth yet. Therefore, only a few of the many ontologies which are published and available online are well documented.

The lack of good documentations makes the reuse process difficult because the decision process of the applicability of a candidate ontology becomes time-consuming. But on the other hand the process of documentation is an additional effort for the ontology developer which still lacks of an appropriate support system. Aiming at creating a support system that allows to understand the subdomains and the coverage of an ontology, we analyzed existing hand-made ontology documentations and identified grouping of concepts as a proper means to provide an overview of an ontology's content and to evaluate its appropriateness. In case of large ontologies with thousands of concepts it is intuitively comprehensible that some kind of complexity reduction is necessary to understand an ontology. For instance, the Friend of a Friend (FOAF) vocabulary[4], which is a small ontology, uses a grouping of concepts in its specification (see Figure 2.7), in order to provide the reader with an easier way to understand the vocabulary.

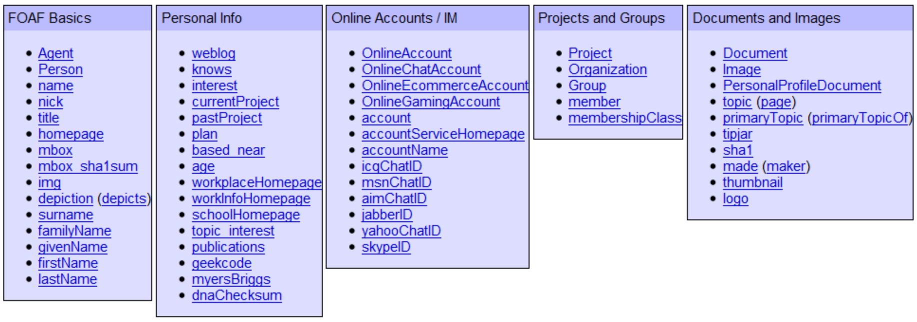
The application of this method for describing an ontology in other documentations like the Music Ontology[5], the Activity Streams Ontology[6], and the Semantic Web Conference Ontology[7] as well as the Vocabulary for biographical

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

[5] http://musicontology.com/

[6] http://xmlns.notu.be/aair/

[7] http://data.semanticweb.org/ns/swc/ontology



FOAF Basics

- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction (depicts)
- surname
- familyName
- givenName
- firstName
- lastName

Personal Info

- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- age
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

Online Accounts / IM

- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- account
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID
- skypeID

Projects and Groups

- Project
- Organization
- Group
- member
- membershipClass

Documents and Images

- Document
- Image
- PersonalProfileDocument
- topic (page)
- primaryTopic (primaryTopicOf)
- tipjar
- sha1
- made (maker)
- thumbnail
- logo

Figure 2.7: Concept groups of the FOAF vocabulary in the specification (version 0.97)

information[8] which are about the same size as FOAF proves how important adequate visualization of meaningful concept groups is. Keeping the rapidly growing Semantic Web [20] in mind and the fact that the large number of ontologies are lightweight and small-sized [18] issues like reusability regarding those ontologies seem to be more urgent. For that reason we analyzed the documentations of the mentioned ontologies and extracted some trends in creating such concept groups. Additionally, we investigated on the applicability of community detection algorithms on the ontology structure in order to identify concept groups automatically or at least semi-automatically. An appropriate concept grouping system is expected to be a useful support for the ontology engineer to create a proper documentation.

Prototyping

For our ontology evaluation analysis we implemented a lightweight web application, which uses R[9] with the igraph[10] library for the implementation of the analysis algorithms. Before identifying the groups, the ontology documents are loaded with Jena[11] and are converted into GraphML files according to the variants which are shown in Table 2.1. Before this process is started the ontologies are loaded in two different ways. Firstly, with inactive inference and secondly, with active inference. Inference has a significant influence on the structure of an ontology.

In order to investigate the applicability of different community detection algorithms to the ontologies, we applied the following algorithms on the different structure variants of the ontologies: Fast Greedy Community (FGC) [14], Walktrap Community (WTC) [45], Spin Glass Community (SGC) [48, 39]. For each these algorithms we created a second version (named WTC_{mod}, SGC_{mod}, FGC_{mod}), that is extended with a weight function for the edges of the graph. This is the first step towards a more semantic-sensitive approach. We use the weights shown in Table 2.2. (The default value for edges which are not listed in the table is 1. If the superclass is *Thing* the *subClassOf* edge value in the table is not used.)

[8] http://vocab.org/bio/0.1/.html

[9] http://www.r-project.org

[10] http://igraph.sourceforge.net

[11] http://jena.sourceforge.net/

Table 2.2: Weights for properties

Property	**Weight**	**Property**	**Weight**
equivalentClass	20	comment	0.2
subClassOf	10	seeAlso	0.2
subPropertyOf	10	isDefinedBy	0.2
domain	5	label	0.2
range	5		

Analysis

As we use hand-made concept grouping to evaluate our results we searched for ontologies, which are divided into concept groups in their documentations. We found the aforementioned ontologies FOAF, MO, AAIR, SWCO and BIO. Figure 2.3 shows the best analysis restuls for the mentioned ontologies.

Table 2.3: Overview of the best results

Ontology	**Best score**	**Graph variant**	**Algorithm**	**Inference state**
FOAF	0.48	V1L, V1LX	FGC$_{mod}$	both(V1L) and off (V1LX)
MO	0.40	V1L	WTC$_{mod}$	off
AAIR	0.88	V3, V3L	WTC$_{mod}$	on
SWCO	1.00	V3, V3L, V3LX	WTC$_{mod}$(V3L,V3LX), FGC$_{mod}$	off
BIO	0.90	V1L	WTC$_{mod}$	on

In case of SWCO it was possible to completely reconstruct the grouping from the documentation. The application of community detection algorithms on ontologies produce good results if concepts are mainly refined with subclasses and the distribution of properties to the classes is balanced. This approach seems to be best to create vertical modules of an ontology which was exactly the expectation, as the main motivation for creating concept groups was to allow an overview on the subdomains. Scores at a low level for FOAF and MO seem to be caused by the characteristics of these vocabularies and their groupings within the documentation. In both cases the central concepts are mainly refined with properties, which is the reason why they contain much more properties than classes and most groups consist of a mixture of properties and classes. After an additional look at the documentations of MO and FOAF (and the latest version of FOAF) the main idea by creating the concept groups in the documentations seem to be the provision of different levels of detail for one domain. This means, that the main goal is to create horizontal modules with different levels of abstraction. Finally, an important observation is that for each ontology the best score was reached with either FGC$_{mod}$ or with WTC$_{mod}$. The introduction of the weight functions improved the results.

2.2.3 Evaluation for Refinement

For an ongoing improvement of an existing ontology it is important to observe its usage. We proposed in [43] to enrich the structure visualization with usage data from the Feedback Tracking and Reporting phases of COLM. By doing so the strength or weakness of the structure as well as the mostly used concepts can be uncovered. This is important to understand how the ontology is used in its context.

Based on our previous work on SONIVIS:Tool and its extension, which is described in [41], Luczak-Rösch and Bischoff present in [34] usage analysis of a Linked Data dataset. After presenting a method for processing the log files of datasets for extracting information about its usage they propose statistic metrics e.g. *Primitive Usage Statistics, Host Statistics.* They utilize different visualization techniques based on the structure of the ontologies to highlight various aspects of the usage (Ontology Heat Map Analysis, Primitive and Resource Usage Analysis, Hosts and Time Analysis, Error Analysis). They are able to detect inconsistencies and make suggestions how to improve the quality of the ontoloy.

Chapter 3

Corporate Semantic Collaboration

In a corporate environment knowledge appears and is used in various situations, e.g., as external information sources, organizational memory, or in automatic processing of events. Semantic technologies (e.g., ontologies and rules) are often the method of choice to represent knowledge. Knowledge emerges by dynamic and collaborative processes within organizations. In order to support these processes including different aspects of collaboration, we decide to focus on chosen problems, concepts and to develop and test our tools for collaborative knowledge processing.

In Section 3.1 we present a short overview over two tools regarding two apsects of the given working package 7: the dynamic access to knowledge through games with a purpose and the derivation and integration of knowledge while detecting trends.

Since concepts of the Semantic Web are not easily be understood by non-experts we believe that the user interfaces has to be designed in a user-friendly way. To evaluate the usability of the One Click Annotator, an editor for creating semantic annotations, we conducted a user study. Furthermore, we are preparing another user study to evaluate different aspects of highlighting annotations. We present both studies in Section 3.2.

3.1 Dynamic Access to Distributed Knowledge (AP 7)

Knowledge can be succesfully shared through collaboration. From a wide spectrum of problems related to knowledge sharing through collaboration, we chose to focus on the ways of accessing different sources of knowledge. In our previous milestone, published in [43], we identify three different sources of knowledge and relevant ways of accessing these sources: access through social networks, access through tagging and access through games with purpose. Among several possibilities of interpretation of distributed knowledge and the problem of dynamic access to it, we defined the working package task as follows: in order to support company workers in knowledge formalization and accessing, we propose an adaptation of extreme tagging concept as a game with purpose. As for the second part of the working package task- trend detection, we consider the access to knowledge as analysis of the web data, i.e. social networks. Regarding relevant problems of knowledge derivation and integration with the goal of detecting trends, we propose to extend the existing knowledge-based trend mining approach [55] into an analysis tool.

In the following Section we describe our extreme tagging tool published in previous reports[41][42] giving an overview over preliminary experiments and relevant issues regarding its extenstion into a game with purpose. We close Section3.1.1 giving an outlook at evaluation. In Section3.1.2 we present shortly the trend mining tool built upon the knowledge-based trend mining method published in[56][57].

3.1.1 Extreme Tagging Tool as a Game with Purpose

The goal of the extreme Tagging Tool for Experts (eXTS) is to help in creating up-to-date user ontologies (see Fig. 3.1) tailored to the expertise fields of the given user group, accessible dynamically and re-usable in different applications. eXTS is a test tool which means that the GUI and the system architecture is only made for the purpose of testing the idea of formalizing experts' knowledge through tag tagging. The idea itself could be realized as an add-on for the browser or a desktop client. However, the GUI of the existing testing tool plays an important role- it should be user friendly, as simple as possible and it should enable the information exchange between its back-end and its user in an intuitive way.

The idea behind the eXTS is as follows:
Based on words from the user's field of expertise (i.e. "computer“, "XML“, "machine“ for computer scientists or "market“, "financial“, "stock“ for financial analyst) a set of initial tags is created (per import from existing sources). These words are presented to the user randomly, triggered by the user through the "Tag“ button. The user's task is to click on a given tag and give any association he or she has on their mind fitting to the word (i.e. "machine“ - "is_kind_of“ - "algorithm“). Additionally, the description of the association is asked to be written before storing it to the database. The back-end functionality ensures the creation of machine-readable description for the words, and concepts their represent, and their associations which can be visualized to the user by using the simple graph visualization and rdf graph visualization.

First version of eXTS

In order to test the formalism of ETS a simple prototype implementing this formalism was developed by the Corporate Semantic Web working group, called eXTS. This prototype is realised as a web application for the Apache Tomcat application server. The front-end is implemented with JSPs and the design rule for the GUI is to keep it as simple as possible. As the implementation of the prototype is not developed for being easy to maintain or extent, in order to integrate the new features a reimplementation seems the easiest way. The functionality of the web services was never used and is not intended to be used in the future, so with the re-implementation the architecture is no longer based

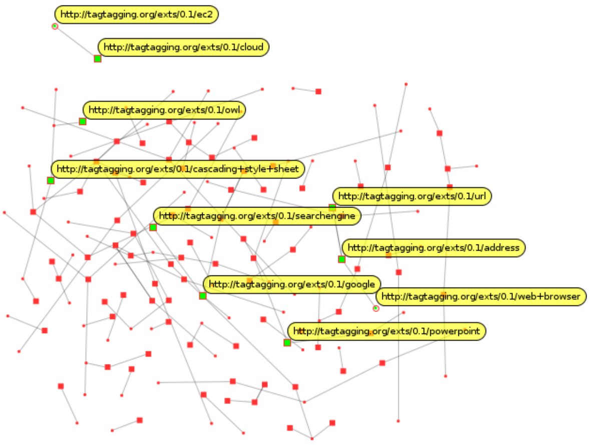

http://tagtagging.org/exts/0.1/ec2
http://tagtagging.org/exts/0.1/cloud
http://tagtagging.org/exts/0.1/owl
http://tagtagging.org/exts/0.1/cascading+style+sheet
http://tagtagging.org/exts/0.1/url
http://tagtagging.org/exts/0.1/searchengine
http://tagtagging.org/exts/0.1/address
http://tagtagging.org/exts/0.1/google
http://tagtagging.org/exts/0.1/web+browser
http://tagtagging.org/exts/0.1/powerpoint

Figure 3.1: A snapshot of the user ontology extracted from the eXTS application

on web services. This simplifies the implementation and reduces the complexity of the server-side architecture.

Second version of eXTS

The second version of the eXTS is (like the first version, too) implemented as a web application. The front-end is realised with JSPs, keeping the same layout as the first version. The back-end is composed of plain java classes and some servlets. The front-end directly accesses methods of the back-end classes. The back-end comprises a real multi-user capability. The user permission handling is done with the Java EE security services and is provided by the application server[1] . This gives an easy way of user management. In a configuration file the server is told, which sites of the application need authorization. If a user visits a site, that needs specific rights, the user is asked to log in. Only if the user logs in with an account that has the required rights to see the site, it is accessible for her. In another configuration file the server is given a database table that contains the registered users and their rights. That is all that needs to be done; the application server handles the rest. Like in the first version the data is stored in a database. But as opposed to the first version, the database is not integrated in the application, but is an external MySQL database. Theback-end communicates with the database with the help of an object-relational mapper. We use Hibernate1 and JPA2 with annotations for this purpose. This allows us to let all the database handling be done automatically by the object-relational mapper and minimizes the need for native SQL queries in the code. For example there is a class (a bean to be precisely) Assignment that represents an assignment a user can create. With the JPA annotation @javax.persistence.Entity this class is marked as an entity class. This gives an easy way of persisting instances of this class to the database and loading instances of this class from the database. To store an instance in the database JPA provides the method persist(). To load an instance from the database the method find() can be used, which is given the

[1] Apacheoranyother

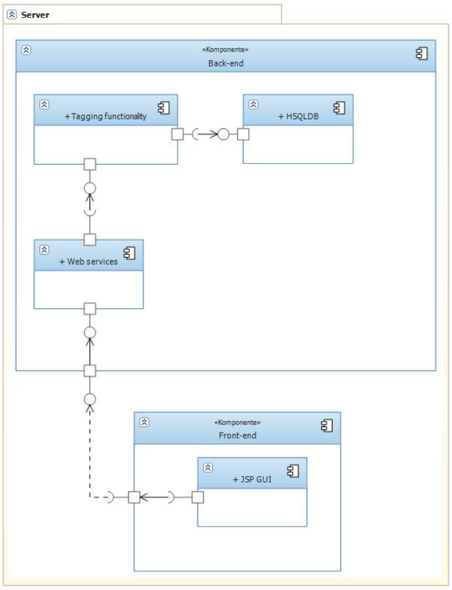
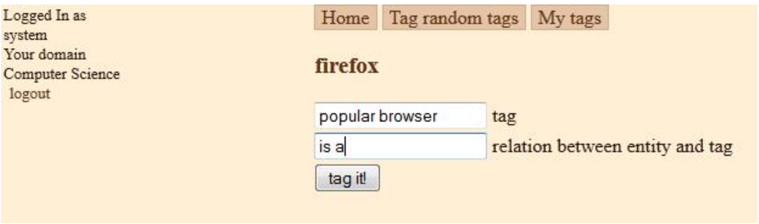


Figure 3.2: UML component diagram gives an overview over the first version of the eXTS system.



Logged In as
system
Your domain
Computer Science
logout

firefox

popular browser tag

is a relation between entity and tag

Figure 3.3: This is a screenshot of the second version of the eXTS implementation, showing the user interface for tagging entities.

class and the primary key of the instance to be loaded and returns a new Java object representing it.

Preliminary results

Below we present some of the preliminary results that were gathered during the first tests:

- A total number of 108 users registered in the system. 59 of these users committed at least one tagging, which corresponds to 54% of the users.
- The users created a total of 1970 assignments.
- For only 273 of those assignments the users chose a relation, which corresponds to 14% of the assignments (depicted in figure 3.4).
- 51% of the users (30 users) only submitted assignments without relations.

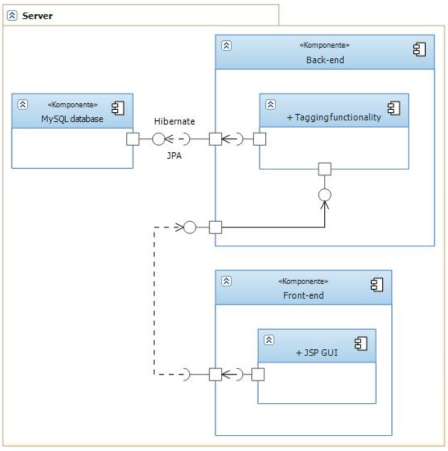


Figure 3.4: UML component diagram gives an overview over the second version of the eXTS system.

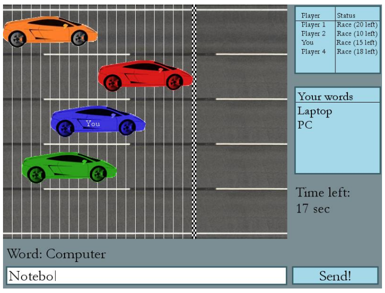
Gaming concept

We decided to extend the eXTS-tool into a game with purpose in order to achieve better results in tool use and a better quality of data generated through it. Our gaming concept is based on three games called: the ”racing game“, the ”falling words game“, and the ”door opening game“ that we present in following paragraphs:

Racing Game The racing game is a multiplayer game that requires at least two players playing it at the same time (at least one of the players must be a human player, the others can be computer controlled players). It is a hybrid game regarding the strategy players follow to reach the winning conditions. This means it is both competitive and collaborative. Competitive here means that every player fights for her own progress in the game to be the first to reach the finishing line. Collaborative means that a player needs to cooperate with other players to win the game, too. The purpose of this game (for me) is to let the players generate new tags for given entities and implicitly strengthen existing entity-tag-pairs. I call an entity-tag-pair being ”implicitlyÂÂ strengthened if it is being generated by a player but was already present in the database before. This means it has a higher weight1 as more than one player produced this pair.

Falling Words Game The falling words game is a singleplayer game, which means it requires only one player. Its purpose is to let the player explicitly strengthen or weaken existing entity-tag-pairs from the database. With ”explicit strengthening or weakening“ (in contrary to the implicit case in the racing game) a player can directly rate an existing entity-tag-pair.

Opening Doors Game The mission for the player is to overcome all three walls and reach the end of the game behind the last wall. She can do this by successfully opening one of the doors in each wall. To open a door the player has to solve different tasks depending on the wall in which the door is. She



Player	Status
Player 1	Race (20 left)
Player 2	Race (10 left)
You	Race (15 left)
Player 4	Race (18 left)

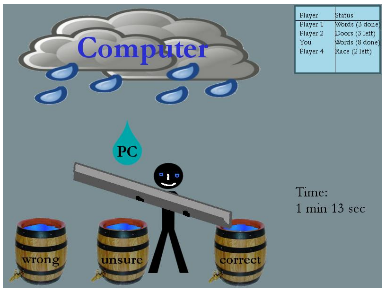
Your words
Laptop
PC

Time left:
17 sec

Word: Computer

Notebo

Figure 3.5: GUI example of racing game

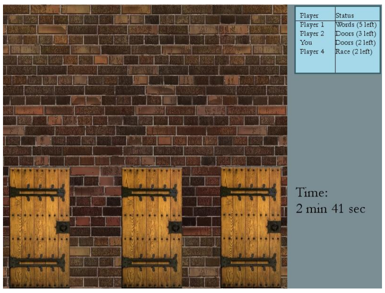


Player	Status
Player 1	Words (3 done)
Player 2	Doors (3 left)
You	Words (8 done)
Player 4	Race (2 left)

Time:
1 min 13 sec

Figure 3.6: GUI example of opening falling words game

can choose one of the three doors she is facing. If she solves the task that is associated with the door, it opens and the player continues to the next wall (or



Player	Status
Player 1	Words (5 left)
Player 2	Doors (3 left)
You	Doors (2 left)
Player 4	Race (2 left)

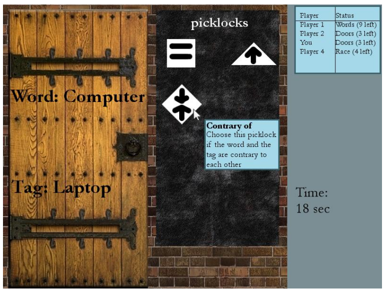
Time:
2 min 41 sec

Figure 3.7: GUI example of opening doors game

to the end of the game if she overcomes the last wall). If the player fails to solve the task, the door becomes blocked and she can choose one of the remaining doors. If all doors are blocked the player has to start all over again, facing the first wall which now contains ”new“ doors.

Outlook

In the previous Sections we gave an overview over the extension of eXTS into a serious game. Tests with the first versions of eXTS as a web application show that users are interested in our tool. Preliminary tests with the first-level game in eXTS are very promising. Currently, further tests are being prepared in order



Word: Computer

Tag: Laptop

Contrary of
Choose this picklock if the word and the tag are contrary to each other

Player	Status
Player 1	Words (9 left)
Player 2	Doors (3 left)
You	Doors (3 left)
Player 4	Race (4 left)

Time:
18 sec

Figure 3.8: GUI example of opening doors game: relations example

to analyse the use of the eXTS tool as a GWAP. We are looking forward to evaluate the final version of eXTS as a GWAP on the one side and to extract useful ontologies from users assignments on the other side.

3.1.2 Trend Mining Tool

As presented in [54][53][57], we propose a knowledge-based method for mining trends. Based on this method, we developed a concept for a trend analysis tool, tremit (see Fig.3.11 for tremit general architecture). Tremit is a sandbox for every researcher, data scientist or developer interested in trend mining in web text data. So far, two state-of-the-art approaches: LDA-based topic modeling and k-means clustering are fully implemented, and our approaches: trend ontology[57] and trend indication[56] are currently being integrated into tremit. Tremit allows for dynamic analysis of an arbitrary text corpus that is formatted in XML and its content is described in German language. The analysis results show trends- the emerging topics- in the texts being analyzed in several different ways. The GUI is a simple Java-Swing GUI that is easy to extend by new functionalities.

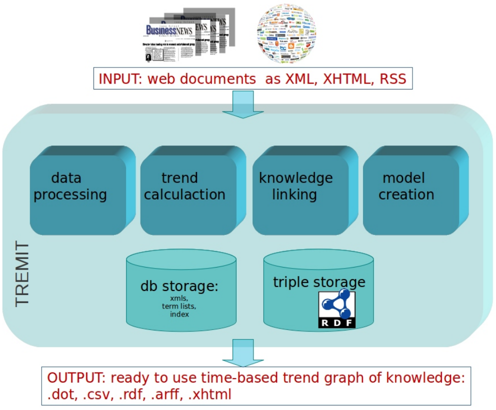
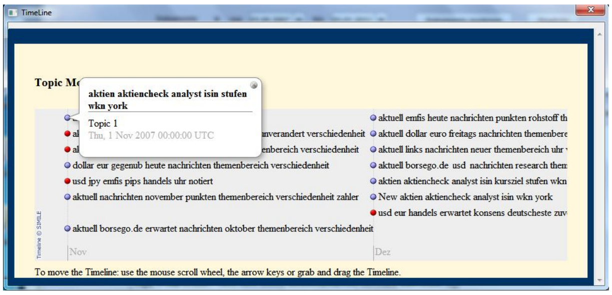

INPUT: web documents as XML, XHTML, RSS
TREMIT
data processing
trend calculaction
knowledge linking
model creation
db storage:
xmls, term lists, index
triple storage
R D F
OUTPUT: ready to use time-based trend graph of knowledge: .dot, .csv, .rdf, .arff, .xhtml

Figure 3.9: A general architecture of trend mining tool



TimeLine

Topic Mo

aktien aktiencheck analyst isin stufen wkn york

Topic 1

Thu, 1 Nov 2007 00:00:00 UTC

● al unverandert verschiedenheit

● al enbereich verschiedenheit

● dollar eur gegenub heute nachrichten themenbereich verschiedenheit

● usd jpy emfis pips handels uhr notiert

● aktuell nachrichten november punkten themenbereich verschiedenheit zahler

● aktuell borsego.de erwartet nachrichten oktober themenbereich verschiedenheit

● aktuell emfis heute nachrichten punkten rohstoff th

● aktuell dollar euro freitags nachrichten themenbere

● aktuell links nachrichten neuer themenbereich uhr

● aktuell borsego.de usd nachrichten research them

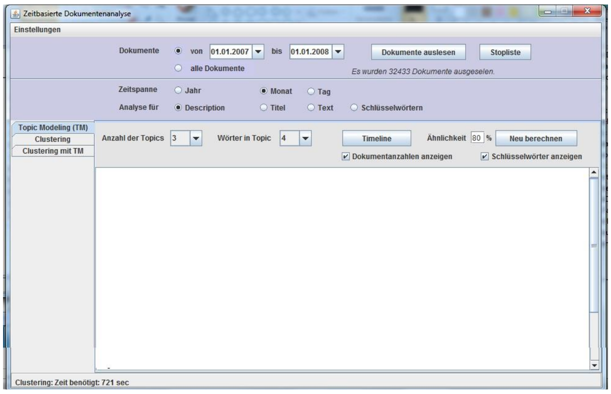
● aktien aktiencheck analyst isin kursziel stufen wkn

● New aktien aktiencheck analyst isin wkn york

● usd eur handels erwartet konsens deutscheste zuv

To move the Timeline: use the mouse scroll wheel, the arrow keys or grab and drag the Timeline.

Figure 3.10: A depict of tremit: topic modeling visualization on test corpus



Zeitbasierte Dokumentenanalyse

Einstellungen

Dokumente ● von 01.01.2007 bis 01.01.2008

○ alle Dokumente

Es wurden 32433 Dokumente ausgeselen.

Zeitspanne ○ Jahr ● Monat ○ Tag

Analyse für ● Description ○ Titel ○ Text ○ Schlüsselwörtern

Anzahl der Topics 3

Wörter in Topic 4

Ähnlichkeit 80 %

☑ Dokumentanzahlen anzeigen

☑ Schlüsselwörter anzeigen

Clustering: Zeit benötigt: 721 sec

Figure 3.11: Tremit GUI

3.1.3 Conclusion

In this Section we described the prototypes for eXTS-tool as a GWAP and the trend analysis tool based on the knowledge-based perspective on trend mining. In this stage of our work, the preparation for the final evaluation of our concepts and tools are ongoing.

3.2 Ontology and Knowledge Evolution through Collaborative Work (AP 8)

In our last report [43] we concluded that current methodologies and tools are rarely found in a corporate environment. In our opinion the complexity of user interfaces (UI) and the necessity of understanding semantic technologies is a main barrier for their dissemination. Semantic technologies should integrate into the working environment seamlessly and adopt existing procedures of user interaction. In our opinion it is essential that all people participating in the generation and the evolution of knowledge can understand and extend the knowledge base without difficulties. In our work we focus on the aspect of collaboratively annotating content resulting in enriched content and background knowledge. Both the content and the knowledge form the basis of semantic applications offering value-added content.

In the past we developed the One Click Annotator (OCA) for creating semantically rich content easily [25]. The OCA is a simple text editor allowing users to select some text describing an entity, e.g., a named entity, and assign the URI of the corresponding resources to it. It interacts with a server component, loomp, which is responsible for managing the annotated content.

Besides the requirement that it should ease the time-consuming task of annotating content, the key challenge in designing the UI of the OCA is gap between the human mindset and the RDF data model. According to the data model facts about a resource is represented as a set of statements without any order. People in contrast are accustomed to an ordered representation of information, e.g., properties are grouped according to their semantics and values are arranged alphabetically or ordered by date. In addition, resources are uniquely identified by URIs allowing machines to distinguish homonym terms. Although people may recognize URLs as unique addresses of websites, they are not used to the idea of identifying entities of the real world with these and are not familiar with the concept of namespaces. Instead, they use labels to refer to them and disambiguate their meaning by their contexts.

To close this gap we followed the example of current word processors and chose well-known UI elements and procedures of user interaction (e.g., formatting a selected text italic) for implementing the annotation process. In a word processor users can choose between different sets of style sheets. Having chosen a set they can select some text and assign a style sheet to it, e.g., heading 1. In the OCA users can similarly choose between vocabularies and assign annotations to text passages.

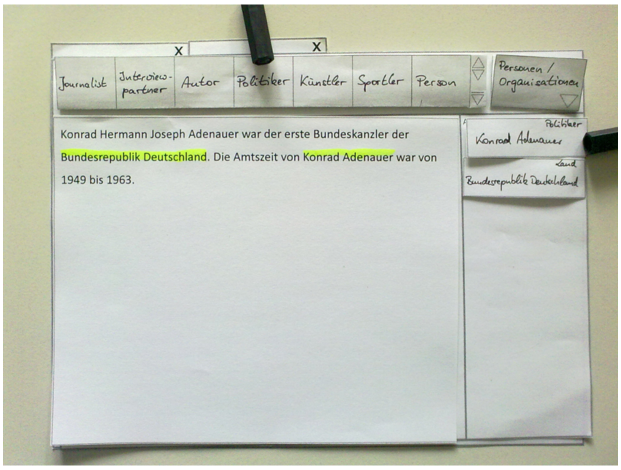
Besides the annotation process we identified the visualization of annotations as another issue of the UI design in the OCA. Since a user can access several vocabularies with the OCA it may be useful to highlight each of them differently, e.g., different colors. Moreover, the editor has to show the relationship between an annotated phrase and the corresponding resource. The task of providing a clear visualization gets even more challenging if we consider annotations spanning a few lines and annotations that overlap each other.

In the following we present our work on the two aforementioned challenges. In Section 3.2.1 we present the results of a user study that we conducted to evaluate the user interface of the OCA. Afterwards, we describe our ideas of a user study for evaluating the visualization of annotations (Section 3.2.2).

3.2.1 User study: One Click Annotator

For evaluating the usability of the OCA we prepared a paper prototype resembling its user interface (Figure 3.12). Paper prototyping is a widely-used method in user-centered design for identifying user's expectations and needs. It has several advantages compared to an implemented user interface: Since paper prototypes are inexpensive to create and to modify, we could easily react on unexpected behavior of users and explore further the user's expectations. Moreover, users feel more comfortable with a mock up and interact with the UI more freely because it does not have the polished look.

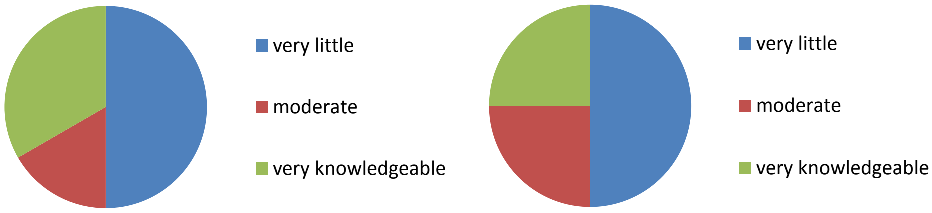
The OCA addresses people that are computer literate (e.g., they know how to work with word processors) and have no or little knowledge in semantic technologies. We invited twelve people matching our requirements most probably to participate in the user study. Figure 3.13 presents the result of the question



Konrad Hermann Joseph Adenauer war der erste Bundeskanzler der Bundesrepublik Deutschland. Die Amtszeit von Konrad Adenauer war von 1949 bis 1963.

Figure 3.12: Paper prototype of the user interface of the OCA

how the participants would themselves assess their knowledge about computers and the Semantic Web.



(a) Computer knowledge

(b) Semantic Web knowledge

Figure 3.13: Computer and Semantic Web knowledge of participants

A study consistent of the following phases: First, we described the purpose of semantic annotations with small examples without showing the OCA itself. Then, we showed the paper prototype to the participant who had time to explore the user interface using a short example text (shown in Figure 3.12); especially, we did not explain the user interface. We prepared the text carefully to lead the participant to anticipate the goals of semantic annotations. Afterwards, we asked the participant to annotate important phrases in a longer text. Although the text was taken from the news domain and, thus, could easily be understood by all participants, it was well chosen and slightly modified to contained some special cases testing the understanding of semantic annotations by the participant. Finally, we interviewed the participant according to a previously defined questionnaire.

Following the user study we evaluated the quality of the annotations by comparing the annotated texts with a gold standard defined by us. As you can see in Figure 3.14 the quality of the results varies. Although half of the participants annotated almost all phrases that we expected two of them created also many ineffective annotations. We consider an annotation as ineffective if it does not refer to a named entity, e.g., a date is not an effective semantic annotation. The diagram also shows that half the participants failed completely in annotating the text.

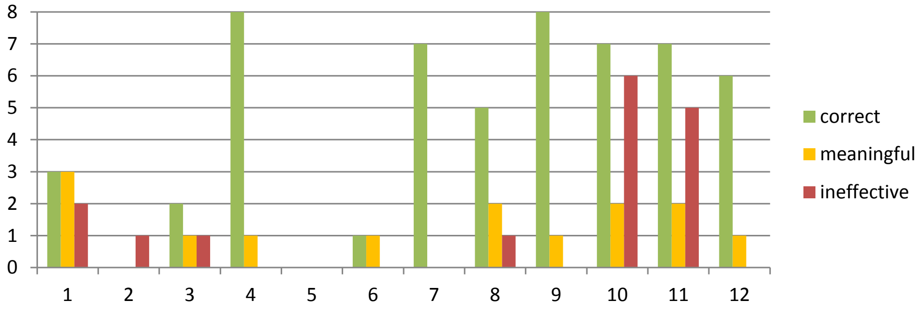


Figure 3.14: Evaluation of the result quality

To understand why so many participants failed we created a diagram about how participants thought about the procedure of creating annotations. Figure 3.15 shows that the user interface itself was well understood by almost all participants. However, they found it difficult to choose the right annotation. Therefore, we believe that the concept of creating semantic annotations and linking named entities to resource was difficult to understand.

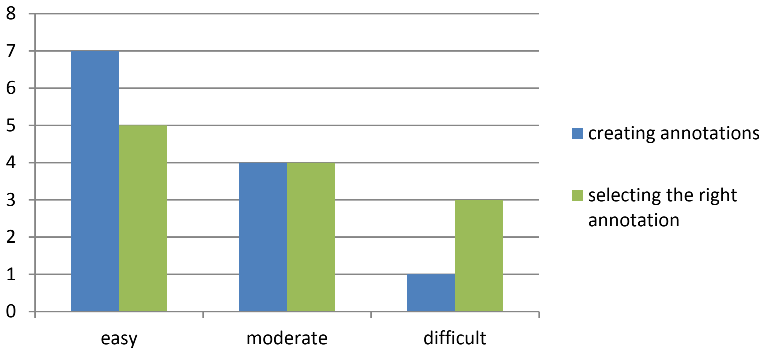


Figure 3.15: Difficulty of annotating with the One Click Annotator

As a result of the user study we believe that the user interface of the One Click annotator is suitable for creating semantic annotations. But although all people, e.g., non-experts, could easily interact with it the process of creating annotations and selecting a resource is difficult. In our future work we will examine what information users need to annotate texts effectively.

3.2.2 Visualizing annotations

Developing the One Click Annotator we were confronted with the task to highlight annotations. Because users already encounter annotated texts in many situations that are not related to semantic technologies, e.g., review mode and comments in word processors, we searched for guidelines describing best practices and user studies evaluating visualizations of annotations. As a result of our search we found many tools supporting text annotations and implementing there own approach of highlighting them. However, we could not find guidelines or user studies as we had expected. It seems that researchers invested up to now little effort in evaluating the question which visualization techniques are most suitable to present and layout annotated texts. In the following we first introduce the problem in greater detail and then describe our ideas for conducting a user study on this subject. The user study forms the basis for evaluating options for highlighting annotaions in a collaborative environment. When knowing how annotations are best visualized in a single user environment we extend our ideas to support concurrent multiuser environments.

In our context we understand annotations as additional information to content, e.g., texts or multimedia objects. These annotations are often embedded into the content using microformats[2] such as RDFa, hCard, or XFN. An annotation typically consists of the two parts atom and annotation (cf. [40]). An *atom* is an elementary continuous piece of information, e.g., a continuous portion of a text or an area of an image. Because the One Click Annotator handles only texts at the moment the term atom will only refer to text atoms in the following. An annotation is contains some additional information that is connected to an atom and is typically created by a third person. Annotations and content do not need to be represented in the same data format.

In the following we compiled a list of properties that can be used for characterizing annotations and the relationships between two of them. All of them influence the visualization of atoms and annotations directly.

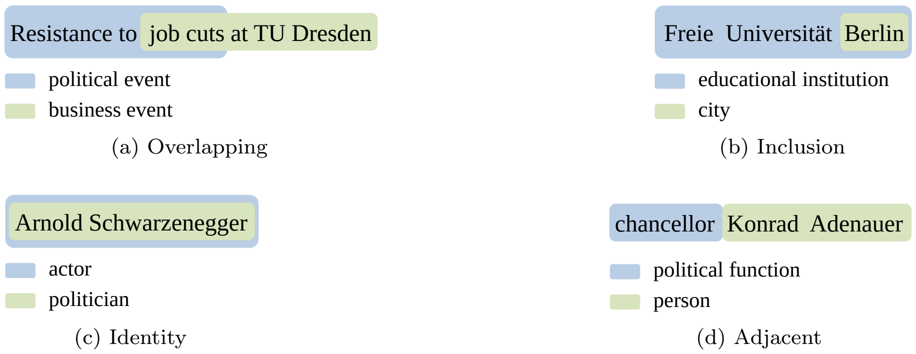
Cardinality. It describes the relationship between atoms and annotations. We differentiate between 1:1, 1:n, n:1, and n:m relationships. In the OCA, for example, we typically have n:1 relationships meaning that several atoms are connected to the same annotation.

Granularity. The application domain may restrict the smallest size of a meaningful atom. Taking text atoms as an example, the smallest size of an atom could be defined as character, word, sentence, sections, and so on. In the context of the OCA we chose words as the size of the smallest atom. If the smallest atoms are whole documents then we often refer to annotations as metadata.

Overlapping. This property describes the positional relationship between two annotations (cf. Figure 3.16). We can basically distinguish between overlapping and adjacent annotations. In the overlapping case two annotated atoms share some parts (e.g., words) with each other. As special forms of overlapping we differentiate between inclusion, an atom is completely contained the other, and identity, two atoms cover the same piece of information but have different annotations. In the adjacent case two annotated atoms are located side by side within the text. Although they are not overlapping at all it is an interesting case of visualizing annotations.

Existing systems focus in general only on the visualization of certain combinations of the above properties. We analyzed some tools for annotating texts with respect to the above properties. The results are listed in Table3.1. We also ranked the visualization of overlapping annotations and their usability: '+'

[2] http://microformats.org/



(a) Overlapping

(b) Inclusion

(c) Identity

(d) Adjacent

Figure 3.16: All variants of overlapping text atoms.

means that the software supports this kind of overlapping completely, '○' indicates that the visualization works out in most cases, and '−' although this kind of overlapping may occur the software does not provide a special visualization.

Property	*Booktate*	*A.nnotate*	*GATE*	*Crocodoc*	*diingo*	*Biblereader*	*veeeb*	*Atlas.ti*	*OpenCalais*	*OCA*
Cardinality										
1:1		×								
1:n	×			×	×	×				
n:1										×
n:m			×				×	×	×	
Granality										
character			×	×	×	×		×		×
word	×	×					×		×	
Overlapping										
overlapping	+	○	+	−		+	+	○	○	
inclusion	+	○	+	−		+	+	○	○	
identity	+		+	−		+	−	○	○	
adjacent	−	−	−	−	+	+	+	○	○	−

Table 3.1: Tools for annotating texts and their supported properties

As an example we describe Booktate[3] in more detail. Booktate is a software for annotating eBooks and transfers processes of paper-based annotations to the Web, e.g., notes on the margin, marking with a highlighter, and underlining. Especially, they invested some effort to support overlapping annotations. An annotation is typically set on word level, however, annotations on section level are also possible. Users can create several annotations for a single atom, thus, it

[3] booktate.com

supports 1:n relationships between atoms and annotations. Atoms are highlighted by assigning a background color or underlining them. An annotation is placed on the margin directly beside the paragraph containing the corresponding atoms and has the same background color as the background color of the atom. If there is not enough space beside the paragraph then the software adds some space below the paragraph. As a consequence a larger empty space may be created between two paragraphs if the first one contains many annotations. To visualize the overlapping of two atoms a subtractive mixture of colors is used. If users hover with their mouse over such an annotation then the original color is restored.

Analyzing theses tools we categorized the concepts of visualizing annotations as follows:

Highlighting. To support users to recognize annotated portions of a text tools highlight atoms. On the on hand tools modify the text style of atoms, e.g., underline and bold. On the other hand they use graphical elements, e.g., background color, frames, and icons. Tools use typically boxes or speech bubbles to represent annotations belonging to some atoms. The background color of the boxes and bubbles corresponds to the one of the related atom.

Position of annotations. There are only a few options to position an annotation: as an overlay near the corresponding atom (sometimes freely movable), on the left or right margin besides the annotated line of paragraph, or below the document. The order of annotations correlates to the order of the atoms in the text.

Overlapping atoms. Since most tools allocate different background colors for each category it bets difficult to present overlapping atoms. We found the following concepts: stripes, mixture of colors, stack view, and vertical lines on the margin. Some of these concepts require a short explanation (cf. Figure 3.17). Mixing of colors means that the overlapping part of two atoms is presented in a color that is derived from them, e.g., the overlapping part of a red and a green atom is shown in a brownish color. The stack view is similar to underlining atoms. While tools underline in black color the stack view adds horizontal lines in the color of the atom below the line.

Connecting atoms and annotations. To illustrate the relationship between atoms and annotations tools assign the same background color to related atoms and annotations, position them nearby, or use mouse over effects.

Resistance to job cuts at TU Dresden
Resistance to job cuts at TU Dresden

(a) Mixture of colors

(b) Stack view

Figure 3.17: Presenting overlapping atoms.

Based on the findings of analyzing tools for annotating texts we are currently developing two concepts for highlighting annotations. These concepts will be evaluated in a user study which is currently prepared.

Chapter 4

Corporate Semantic Search

Whereas previously presented research fields of corporate ontology engineering and corporate semantic collaboration focus on different aspects of capturing and formalizing enterprise and domain knowledge in form of ontologies, this chapter concentrates on methods for providing users within the corporate context, both internal and external, with personalized context-aware access to enterprise data.

In section 4.1 we tackle the problem of searching and extracting information from non-textual data. We observe that due to the visual and audible nature of multimedia content, it proves extremely hard for machines to understand this information and process it in a meaningful way. We conclude that collaborative approaches are needed that combine automatic information extraction with human aided annotation. We discuss the shortcomings of existing semantic annotation systems and describe a multimedia annotation approach that focuses on a combination of Semantic Web, croudsourcing and machine learning techniques in order to overcome the difficulty that machines have in understanding multimedia content.

In Section 4.2 we focus on the application of the Semantic Web technologies in the field of recommender systems. We describe a domain independent similarity measure for recommender systems based on the idea of property propagation, we discuss the main shortcomings of this approach and show how it can be improved as well as describe its integration into the Semantic Matchmaking Framework presented in the first phase of our project.

In Section 4.3. we generalize our solution to capture the various aspects of semantic search including personalized and contextual access to heterogeneous data sources. On the basis of the W3C Linked Data approach we depict our proposal for a general ontology model for representing context information in a flexible and extensible way, which can be used for personalized situation-aware semantic search and information access.

4.1 Searching Non-Textual Data (AP3)

Multimedia content has become one of the most important type of resources available on the World Wide Web, however our understanding of it is severely limited due to its non-textual nature. In order to overcome this problem, large

Web 2.0 sites such as Flickr[1] and Youtube[2] allow their users to assign free-text tags multimedia content such as images respectively videos, in order to better index and retrieve it. However, due to its arbitrary nature this approach is limited in it's applicability in scenarios that require machine processing of annotations. The main drawbacks are the lack of a consensual controlled vocabulary for tagging [27], lack of a standardized mechanism for granular annotation of multimedia content and lack of reusability[60] and lack of support mechanisms based on machine learning in order to support the user in the task of manual annotation[52]. In [43] we introduced a state of the art analysis of different Semantic Web Ontologies that try to tackle the problem of multimedia annotation at a structural as well as semantical level. We also analysed different existing metadata formats for multimedia and how they can be incorporated into ontology-based annotations.

In order to be able to better understand the semantics of multimedia and thus be able to better retrieve and monetize this content, we observe the need innovative and intuitive annotation tools. In this section we propose an architecture for the annotation and retrieval of multimedia content that makes use of Semantic Web technologies such as Ontologies and Linked Data as well as croudsourcing[28] and machine learning.

4.1.1 Multimedia Annotation

Multimedia annotation is a difficult task and it varies in difficulty based on the different types of multimedia contents existing on the web. Image and Audio annotation require a less effort than video annotation that combines the previous 2 categories. Due to its complexity and the large amount of video material on the web that we will focus in our work on this type of content. However, the same techniques can be modified to work for image and audio content.

Annotation Types

The first task in building an annotation tool is to understand what types of metadata annotations it needs to handle. According to [49] we can divide video annotations into 3 categories

- Bibliographic annotations which are used to describe information related to the video such as title, creation date, description and genre as well as information related to the people involved in the video such as producer, director, cast etc.
- Structural annotations that deal with the low-level technical attributes of the video such as segments, scenes and shots
- Content annotations such as annotations of the objects and persons in each specific scene in the video and general keywords associated with the video.

However for our purposes we revise this division into 2 main categories:

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

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

- *Semantic structure annotations*: which try to represent the low level features of video content using semantic web technologies such as the ontologies presented in our previous report

- *Semantic content annotations*: which include content annotations as well as bibliographic annotations and represent them using external domain ontologies such as Dublin Core[3], FOAF[4] or DBpedia[5].

In order to be able to efficiently annotate multimedia content our tool needs to support both type of semantic video annotations. Structural annotations are important because they provide the "semantic glue" necessary to bind the content annotations to the specific scene or key-frame in the video sequence. Furthermore they provide us with the necessary overall format for the representation and storage of the annotations. Content annotations provide the real semantics behind the annotations and allow us to make use of the abundancy of existing external Domain Ontologies as well as the linked data resources available in the Linked Data Cloud[6].

Previous Work

A lot of work has gone into creating various tools for semantic multimedia annotation tools. A fairly recent study [19] surveys 19 different semantic image video annotation tols. Some of the most advanced of these tools are:

- The Video and Image Annotation Tool (VIA)[7], which enables granular semantic structure annotations based on MPEG-7 as well as the semantic content annotations based on imported owl files. Furthermore the resulting annotations can be exported in the RDF format

- The IBM VideoAnnEx[8] annotation tool is similar to VIA in that it is based on MPEG-7 for semantic structure annotations as well as semantic content annotations based on imported XML vocabularies. However it is ill suited for the integration into the Semantic Web since it does not support the basic data model of RDF and its development has been halted.

- The Ontolog[9] annotation tool differs from the previous tools in that it supports semantic structure annotations based on its own simplified schema rather than on an MPEG-7 derived ontology. Furthermore, it enables semantic content annotations trough the inclusion of multiple external domain ontologies as well as the creation ad-hoc ontologies.

- The K-Space Annotation Tool (KAT)[10] is particularly interesting because in contrast to other annotation tools that use the MPEG-7 XML vocabulary or MPEG-7 inspired OWL translations, KAT employs the Core Ontology for Multimedia (COMM). COMM tries to create a modular ontologic

[3] `http://dublincore.org/`

[4] `http://www.foaf-project.org/`

[5] `http://dbpedia.org/About`

[6] `http://linkeddata.org/`

[7] `http://mklab.iti.gr/project/via`

[8] `http://www.research.ibm.com/VideoAnnEx/index.html`

[9] `http://www.idi.ntnu.no/heggland/ontolog/`

[10] `htpps://launchpad.net/kat`

framework for the representation of semantic structure annotations. Due to its ontological framework it can be argued that the structural annotations provided by KAT are semantically richer than other existing tools. In addition, KAT provides import functions for external domain ontologies as well as the feature to export the resulting annotations in the RDF format.

4.1.2 Semantic Annotation Framework for Multimedia

Requirements

While testing the above mentioned tools and analysing others, we came to the conclusion that while most of the tools provide decent annotation functionality for both semantic structure and content annotations. However, we have noted a series of major drawbacks in all the existing multimedia annotation tools. These tools require users to manually import OWL or XML files in order to be able to use ontology resources, there is no mechanism for making use of external Linked Data Resources other than by creating custom files and importing them by hand. In addition, due to their relative early creation date and the fact that these tools have not been updated in years, they do not make use of the most recently adopted standards in the Semantic Web community. Furthermore, most of these tools are based on desktop applications and offer next to no facilities for collaboration between users. Another important drawback is that, none of them offer any machine learning features that can assist the user in the difficult and time-consuming task of manual annotation, requiring large amounts of extra work for tasks that can be easily handled by basic machine learning frameworks.

In order to develop an annotation tool that can efficiently annotate multimedia resources and concluding from our previous observations of existing tools we drafted the following list of requirements:

- Web-based annotations: Due to the fact that most if not all large multimedia portals such as Youtube, Facebook[11] or Flickr offer their users web-based annotations we not a definite fammiliarity with such systems. It would be hard to convince users to download a desktop application to achieve the same feat.

- Suggestion of Ontology-based tags: one of the major drawbacks of the current annotation systems is the lack of controlled vocabularies or more semantically rich alternatives such as ontologies. Users type tags based on their current thoughts, these tags in turn can differ in syntax, language and semantics. By providing tag suggestions to the user based on external domain ontologies and Linked Data resources we can assure a consistent and machine-understandable annotation process.

- User collaboration: creating a comprehensive semantic description trough tagging, even with Ontology-based tags, is a time consuming and difficult task. No single user can achieve this efficiently. Therefore we need to implement a collaboration mechanism that allows multiple users to annotate the same resource and a curation mechanism for detecting duplicates and/or semantically related tags.

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

- Deep annotations: current Web 2.0 systems focus largely on tagging with the purpose of better indexing and retrieving entire files. Trough fine-grained scene and keyframe level annotations we can retrieve fragments and not only the entire file

- Use of current standards: In the past year the W3C developed 2 new standards for semantic multimedia, namely the Ontology for Multimedia Resources and the Media Fragments Uri Specification[60]. These standards allow us to annotate multimedia resources as a whole as well as fine-grained annotations. Furthermore they provide us with a standardized exchange format and allow us to be compatible with other future applications that may use the same standards.

- Integration in the Linked Data Cloud: we can improve the efficiency of our annotations 2 fold by making use of existing linked data resources such as DBpedia.org or multilingual versions of DBpedia such as the German DBpedia [12]. Trough the use of customized SPARQL queries we can suggest ontology resources as well as linked data resource for annotation purposes. Furthermore by the resulting semantic associations between the multimedia files and these resources we can greatly improve the precision and recall of multimedia searches. Another benefit of our approach is the augmentation and improvement of the current Linked Data Cloud by interlinking our resulting semantic descriptions of multimedia files with existing hubs such as DBpedia.org or the German DBpedia which we administer.

- Integration of machine learning techniques: Video processing and indexing is a field of intense research in the Artificial Intelligence community. The amount of work invested in object recognition and object tracking is staggering and the last years have yielded particularly interesting results which open a series of new possibilities for multimedia annotation. By using existing algorithms for the extraction of shots or the tracking of objects we can ease the burden on the shoulders of users and make manual annotation a more easy task.

Architecture

The architecture of the system can be best understood by following the following workflow which is executed when a user wants to annotate a multimedia item. The user accesses the Annotation GUI where he is presented with multimedia items he can annotate. The items to be annotated will generally be presented as a single image in the case where we want to annotate images or a series of shots extracted from a video. The shots or key-frames have been previously extracted by the Multimedia Preprocessor component of the Semantic Annotation Service. When a user wants to annotate something he can draw an are of interest in the video that contains the object of interest he wishes to annotate. When the user has selected his object of interest, the Semantic Annotation Recommender kicks into action. The user can select from a series of predefined annotations based on different domains of interest, or simply type in a free-text annotation that will then be compared with existing Ontologies and Concepts from those ontologies will be suggested to the user based on the typed in text. To further enhance the

[12] `http://de.dbpedia.org/`

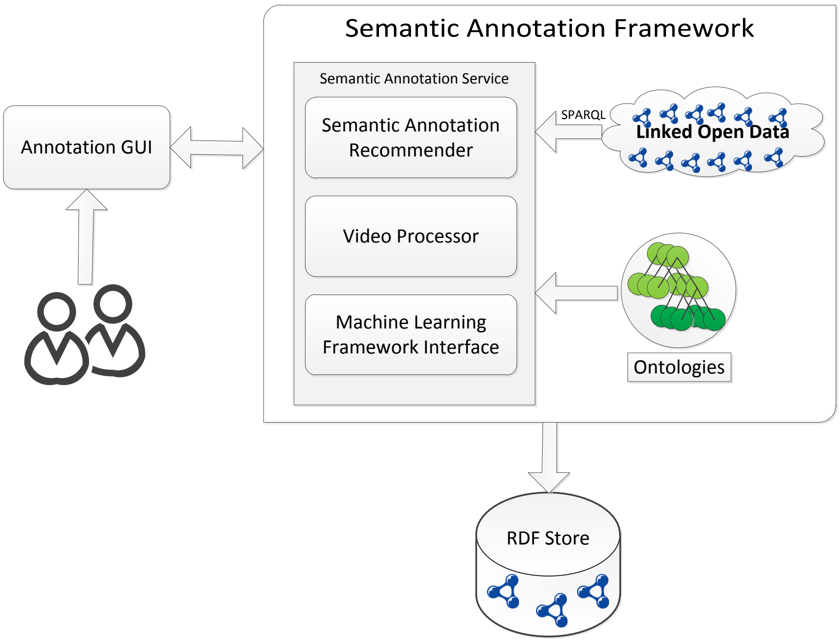

Semantic Annotation Framework
Semantic Annotation Service
Semantic Annotation Recommender
Video Processor
Machine Learning Framework Interface
Annotation GUI
SPARQL
Linked Open Data
Ontologies
RDF Store

Figure 4.1: Architecture of the Semantic Annotation Framework for Multimedia

relevance of annotations, not only ontology concepts will be suggested, but the Semantic Annotation Recommender (SAR) will search for matching concepts from the Linked Open Data Cloud (LOD) and suggest them as an option for annotation to the user. In order to further enhance the annotation experience of the user, we will try to ease certain tasks trough the integration of Machine Learning Frameworks that can increase the ease of use of the system as well as the quality of the annotations. For example, when a user annotates an object, the user does not have to track the object and annotate it in every key-frame, the machine learning component will track and annotate the object in all previous and following frames for the user. Another important task of the Annotation GUI is to allow the user to perform granular annotation of entire fragments in movies. The user will be able to specify when a scene starts and ends as well as asigning it meaningfull anotations. The system will then generate a unique URI for that fragment based on the Media Fradments URI specification. Furthermore, trough the interaction between the Annotation GUI and the Semantic Annotation Service the user will be able to create complete semantic structure annotation of multimedia items based on the latest W3C Ontology for Media Resource standard [31]

4.1.3 Conclusion

In this section we proposed an innovative approach for the extraction of relevant information from multimedia data trough the use of semantic annotations. We described a prototype architecture that combined the streangts of semantic web technologies with courdsourcing and machine learning approaches in order to achieve efficient annotation of multimedia data twords the purpose of better information management and retrieval.

4.2 Personalization and Contextualization of Search (AP4, AP 14)

The application of personalization and context-aware search techniques provides the greatest benefits in environments characterized by user diversity with respect to their preferences, knowledge, goals, environmental context, etc. Such conditions can clearly be observed within business enterprises where personalization and contextualization can be targeted at internal (employees) and external (customers or business partners) users [16]. From the business perspective, the most relevant kind of adaptive systems, providing personalized information access, are recommender systems, due to their prevalence in e-commerce applications like online-shops.

Recommender systems address the problem of information overload by reducing the search space to items or resources of interest to the user. In [43] we delivered a state-of-the art analysis of different classic recommendation approaches such as collaborative filtering, content based filtering as well as knowledge based recommender. Furthermore, we discussed various possible improvements which result from the application of Semantic Web technologies as well as referenced some examples of implemented Semantic Web recommender systems. We also addressed the issues regarding user model, which is the central component of every adaptive or adaptable system, and provided an overview of various kinds of user related information including:

- interests/preferences
- knowledge
- background
- user context
  - goals and tasks
  - platform
  - physical context
  - human dimension (personal and social context)

We discussed what role those kinds of user related information may play in the process of generating recommendations as well as described their most common representations in recommender systems. We argued that Semantic Web technologies not only provide recommender systems with a more precise understanding of the application domain, formalized in ontologies, but can also be utilized for a richer representation of user related information, through modeling of user profiles as an overlay of an underlying domain and user context ontology, based on existing conceptualizations, for instance the General User Model and Context Ontology GUMO [24] represented in OWL.

As far as previous research on the application of Semantic Web technologies to the recommendation process is concerned, we can distinguish between

- **hybrid approaches** which combine classic algorithms with semantic extensions, and
- **"fully" semantic recommender systems** which solely relay on ontological data

In the former category, Semantic Web technologies enhance the classic algorithms, for example: by introducing item features into computation of user similarity in collaborative filtering [61], by addressing the problems of feature extraction in content based filtering through formal representations of items to be recommended (e.g. GoodRelations [26]) and user related information (e.g. GUMO [24]), or by integrating missing or additional item features from distributed sources (e.g. Linked Data [8]). In the latter category, the recommendation algorithm has to be adapted, in order to be able to fully utilize the relations between ontology concepts. This can be done by either providing domain specific recommendation rules or by relying on the graph representation of ontologies. In our previous work [41] we presented a domain-dependant approach applied to the task of personalized museum search, in which, for example, a user looking for museums related to a particular artist also gets recommendations of museums related to the art movement of the given artist. In contrast, domain independent approaches calculate item similarity by comparing their graph representations.

In the remainder of this section, we introduce an example of a domain independent similarity measure for recommender systems based on the idea of property propagation, we discuss the main shortcomings of this approach and show how it can be improved, as well as describe its integration into the Semantic Matchmaking Framework presented in the first phase of our project [42].

4.2.1 Semantic Similarity Measure Based on Property Propagation

In the Semantic Web community the research on domain-independent semantic similarity measures has been initially applied to address the challenge of ontology alignment [17] based on string, lexical, and structural matching. This task, however, faces slightly different challenges from those which arise from calculating recommendations. Hence, those measures usually do not suit the recommendation domain and have to be adapted. In the past few years, Semantic Web technologies and recommender systems have enjoyed growing interest and many approaches exploiting semantic information described in ontologies have been proposed in literature (e.g. [3], [9], [37]). The majority of those approaches is domain-dependant and does not fulfill our requirements for a generic, flexible recommender architecture [42].

Recently, Lémdani et al. [32] presented a domain-independent ontology-based similarity measure for recommender systems inspired by the graph matching algorithm introduced in [36]. The main assumption underlying this approach is that an item is defined by its surrounding resources (by means of properties) and those individuals, in turn, are also defined by their neighborhood. In other words, each concept in an ontology is described by its local context. Consequently, the similarity computation of two given items of the same nature (i.e. instances of the same ontology class) is propagated to resources describing those items, in an iterative manner. In the first step, the similarity is initialized so that two identical items have the maximum similarity equal 1, while the similarity between two different items is set to 0.

$$sim_0 = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

As next, the similarity between two items i and j is computed in an iterative way so that for two identical instances the similarity equals 1, otherwise it is calculated as the average of the semantic similarities of the pairs $(i^{'}, j^{'})$ related to (i, j).

$$sim_{k+1}(i,j) = \begin{cases} 1 & \text{if } i = j \\ \sum_{E_{i,j}} \frac{sim_k(i^{'},j^{'})}{|E_{i,j}|} & \text{otherwise} \end{cases} \tag{4.2}$$

$$\text{where } E_{i,j} = \{(i^{'}, j^{'}) | \exists R \in \mathcal{R} R(i, i^{'}) \wedge R(j, j^{'})\}$$

Consequently, for $k = 0$ the semantic similarity $sim_1(i,j)$ only takes into account individuals that are both related to i and j by the same property, due to equation (4.1), whereas for $k > 0$ also instances being at a distance k from i and j influence the similarity of the two items[13]. Since the function sim is convergent, its computation reaches a fixpoint [46].

The described similarity measure can be used as a main component of a semantic recommender system and is flexible enough to be included in other recommendation modules. The advantage of this approach, apart from its domain-independent nature, is, that the similarity computation can be performed off-line and then used in real-time recommendations, which provides a significant performance boost. The evaluation in the domain of research papers and movies showed that the propagation of similarity measure detected similar pairs of items which were not detected by classic similarity measure thereby leading to increased recommendation quality [32].

Through an analysis and prototypical implementation of this approach we have identified three main drawbacks of the similarity measure introduced by Lémdani et al.

- The propagation algorithm only takes *object properties* (i.e. properties representing relations between instances of two classes) into account. Since some ontologies may relay on RDF Literals for description of item features, extending the property propagation algorithm onto *datatype properties* would lead to increased recommendation quality. This can be realized by mapping datatype properites to similarity measures for string Literals (e.g. Jaro-Winkler distance [63]) and other XML Schema datatypes[14]. If, in the process of property propagation, a literal value is encountered the corresponding matcher will be invoked by replacing the equation (4.2) with the pre-defined similarity measure and the property propagation stopped.

- Through our empirical tests we came to the finding, that, in the case of multiple properties of the same kind, the computation of the similarity as the average of all value pairs may lead to counter-intuitive results.

[13] An instance A is at a distance k from B if there is a sequence of k properties linking them together

[14] http://www.w3.org/TR/xmlschema-2/

For instance, consider the genres of two films f_1*(comedy)* and f_2*(comedy, science-fiction)*. In this simple example the similarity equals *0.5*[15] even though both films share the same genre *comedy*. Based on this finding, we argue that given multiple properties of the same kind, it would make more sense to compute the average only if no perfect match was found. This would not only improve the recommendation quality but also have a positive impact on the performance of the algorithm.

- Another characteristic of the described property propagation algorithm is, that all properties have the same impact on the similarity computation. We think, that the recommendation quality of the algorithm can further be improved by introducing a property weighting function. The weights can be initialized for a particular use case scenario, and then adapted to reflect user preferences based on either explicit user input or by learning from user interaction with the recommender system.

4.2.2 Integration into SemMF

The implementation of a domain-specific application architecture supporting personalized search based on user profiles requires a suitable component for ranking of resources with respect to user preferences. The process of finding best alternatives for a given user profile is called matchmaking. Such component should offer application developers a ready-to-use tool allowing a fast implementation of the matchmaking logic, thereby reducing the cost of the overall application development. The key requirements for such a tool are:

- **domain-independent generic architecture**

  being able to handle various corporate resources and user profiles regardless of the underlying data schema (ontology T-Box)

- **flexibility**

  i.e. offer various matchmaking techniques for different kinds of object properties

- **extensibility**

  i.e. provide interfaces for implementation of new (domain specific) matchmaking techniques

- **traceability**

  i.e. deliver a detailed explanation of the matchmaking result together with the similarity ranking

Given these requirements, we designed a flexible Semantic Matchmaking Framework[16] for calculating semantic similarity of multi-attributive and multi-dimensional information objects represented as arbitrary RDF graphs with concepts from an underlying corporate or domain ontology. In the corporate context, such information objects may represent, on the one hand, enterprise resources like products, services, employees, business partners, documents (including metadata), etc. On the other hand, they may represent user profiles.

[15] In this example we leave out the property propagation step for clarity

[16] http://semmf.ag-nbi.de/

In general, the framework can be applied in a wide range of use case scenarios ranging from product/service recommender systems to expert finder systems.

Depending on the type and semantics of object attributes or dimensions the framework should support different kinds of similarity measures, for example:

- **string-based**

  Calculating the similarity of two string values represented by RDF Literals. This includes comparing keywords, searching for keywords (and their synonyms) in texts, searching for Named Entities, or applying Natural Language Processing techniques.

- **numeric**

  Used to determine similarity of two numeric values.

- **taxonomic**

  Applied for matching attribute values represented by resources from a common taxonomy. An example of such taxonomic matcher inspired by the work from [64] is included in SemMF distribution. The similarity between two concepts c_1 and c_2 can be determined based on the distance $d(c_1, c_2)$ between them, which reflects their respective position in the concept hierarchy. Consequently, the concept similarity is defined as: $sim(c_1, c_2) = 1 - d(c_1, c_2)$. For the calculation of the distance between concepts we utilize an exponential function which implies two assumptions: (1) the semantic difference between upper level concepts is greater than between lower level concepts (in other words: two general concepts are less similar than two specialized ones) and (2) that the distance between "brothers" is greater than the distance between "parent" and "child".

- **ontology-based**

  Computing similarity between ontology concepts based on a variety of relations (i.e. not only subsumption properties) defined in a common ontology. In the previous section we described an example of such ontology-based similarity measure.

- **rule-based**

  Which given a set of pre-defined rules determine the similarity between complex object dimensions. Consider, for example, an expert finder scenario in which, while searching for experienced Java programmers, only those candidates would receive a high ranking whose skill matches the concept *Java*, and additionally have already worked in projects for which *Java* skills were required.

- **(geo)location-based**

  For performing vicinity search given two locations (cities, street names, etc.) as strings or geo coordinates.

- **collaborative filtering**

  Taking into account not only a given user profile but also preferences of similar users, with respect to a particular attribute or dimension to be matched.

As depicted in Figure 4.2, the Matchmaking Framework plays a key role in realizing Web applications supporting personalized search in corporate data. In a given use case scenario, through a domain-specific Web interface, users provide their query and preferences which are represented in RDF using concepts from an underlying corporate or domain ontology. Alternatively, a user monitoring component can easily be integrated into SemMF. As next, a user profile is merged with the use-case-specific matchmaking configuration delivered by the application administrator. It includes, among others, the selection and mapping of attributes/dimensions in user profiles with the semantically corresponding attributes/dimensions in corporate information objects to be matched, together with information about which matching techniques should be applied for computation of each attribute/dimension similarity as well as initial property weights (see Section 4.2.1). The aggregated RDF graph is then passed (as query object) to the Matchmaking Engine.

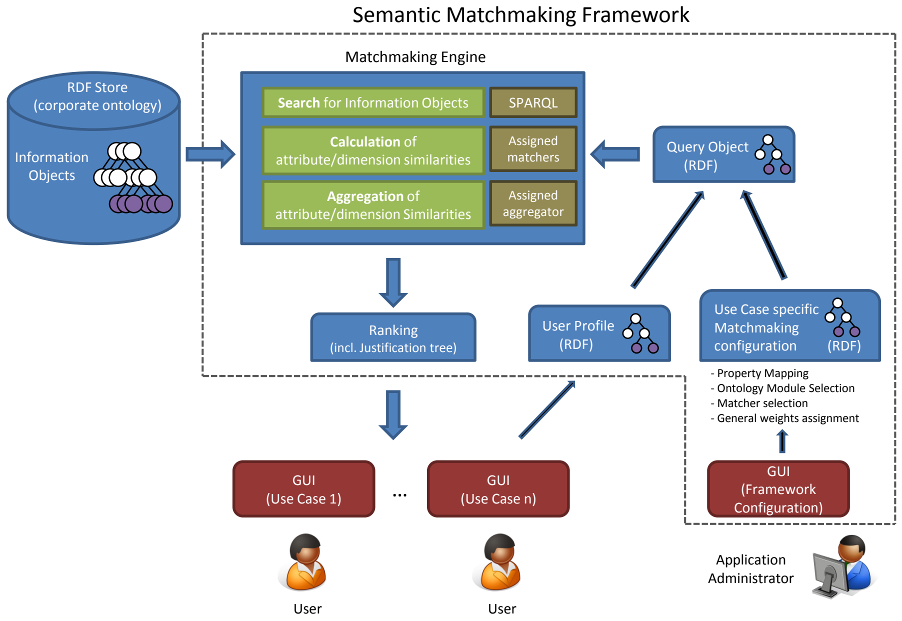


Figure 4.2: Architecture of the Semantic Matchmaking Framework

The process of matchmaking is carried out by the engine in three steps. First, the information objects to be matched, together with all relevant background knowledge (e.g. concept taxonomies), are retrieved from the RDF store. The access to RDF data is realized with *Jena - the Semantic Web Framework for Java* [13]. As next, for each information object, the engine computes the attribute/dimension similarities by invoking appropriate matchers implementing a certain matching technique specified by the application administrator. Finally, all attribute/dimension similarities are aggregated into an overall similarity score for a particular information object. The output of the engine is a ranking of information objects for a given user profile with additional information containing the explanation of the matchmaking process for each single object. The result is rendered in an appropriate format and presented to the user via the application-specific Web interface.

4.2.3 Conclusion

In Section 4.2 we focused on the application of the Semantic Web technologies in the field of recommender systems. We described a domain independent similarity measure for recommender systems based on the idea of property propagation, we discussed the main shortcomings of this approach and showed how it can be improved. We also demonstrated that this adapted similarity measure can easily be integrated into the Semantic Matchmaking Framework as a new matcher.

4.3 Ontological Representation of Context (AP 14)

Since the explosion of information on the Web users are confronted with a huge information overload making it increasingly difficult to find relevant information for knowledge-intensive tasks or to make an optimum choice from vast amounts of alternative resources such as, for example, products or services. This problem is addressed by adaptive and adaptable software systems which aim at providing personalized and context-aware access to huge amounts of information [38]. The application of personalization and context-aware search techniques delivers the greatest benefits in environments characterized by user diversity with respect to their preferences, knowledge, goals, environmental context, etc. Such conditions can clearly be observed within business enterprises where personalization and contextualization can be targeted at internal (employees) and external (customers or business partners) users [16].

In this section we first concentrate on general aspects of semantic search and semantic supported access to heterogenous data sources, followed by a review of issues regarding user modeling, also taking into account various kinds of contextual information. We then depict our ontological model for representing contextual information and user context.

4.3.1 Aspects of Semantic Search on Heterogenous Data Sources

Personalization and contextualization of semantic search solutions comprises various different aspects. Figure 4.3 shows a conceptual classification model for typical search aspects.

The search repositories in corporate environments often consist of heterogeneous data from unstructured text data to semi-structured XML and structured relational data in databases as well as multi-modal and multi-media data. We follow the W3C Linked Data approach for exposing, sharing, and connecting pieces of these data, information, and knowledge on the (Corporate) Semantic Web using URIs and RDF. The four main design principles of the Linked Data principle are:

- Use URIs to identify things.
- Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.

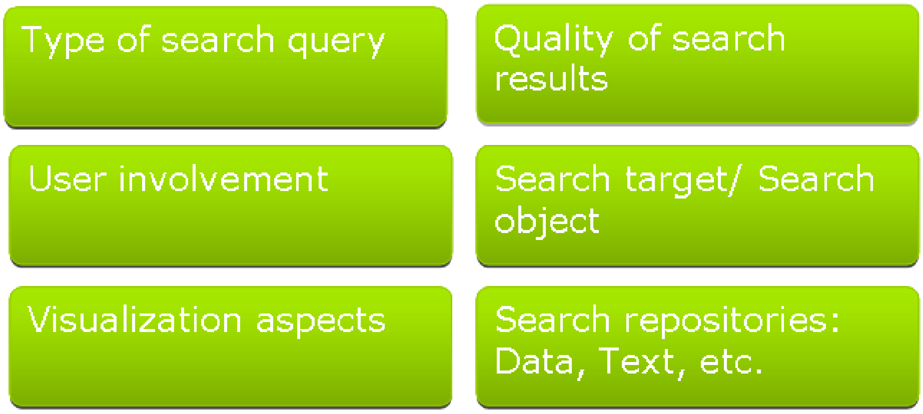

Type of search query
Quality of search results
User involvement
Search target/ Search object
Visualization aspects
Search repositories: Data, Text, etc.

Figure 4.3: Search Aspects

- Provide useful information about the thing when its URI is dereferenced, using standard formats such as RDF/XML.
- Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.

There are tools available which allow extracting and translating structured, semi-structured and unstructured data into the RDF data model so that it can be linked with other Linked Data resources and published on the Web. A particular aspect in providing meaningful interpretation of this data in terms of information is the linking with background knowledge such as ontologies so that semantic interpreters can understand (interpret) this information and search for it semantically. This approach is used in the growing Linked Open Data Cloud, e.g. in the Linked Data version of Wikipedias' data, called DBpedia, and its language specific versions such as DBPedia Deutsch.

In the previous section we have described a semantic search approach for personalized recommender systems which exploit semantic similarity matching techniques. In the following we first review common aspects of user and context modeling and then depict a general ontological representation model for representing context information.

4.3.2 User and Context Modeling

One of the central components of every adaptive or adaptable system is the user model. It represents information about individual users required by the system in order to provide the adaptation effect [12]. The process of creating and maintaining the user model is referred in the literature as user modeling. Depending on the information being modelled, we can identify models representing features of users and models that are rather concerned with the context of the user's work or search scenario. Typical features and context related information relevant in user modeling are [12]:

- **Interests/Preferences** are, in general, the most important, and in most cases the only, part of a user model in adaptive information retrieval and filtering systems as well as in (content-based) Web recommender systems in particular, where they are referred to as user profiles. The most common representation of user profiles, still up to this day, is a weighted vector of keywords extracted from textual data [1]. In contrast to this approach, concept-based user profiles represent user interests as an (weighted) overlay of a domain model, for example in form of an ontology [50]. Concept-based models are generally more powerful than keyword-based models due to their ability to represent user interests in a more accurate way (thus avoiding common problems associated with term ambiguity). Additionally, semantic links in the underlying domain ontology enable interest propagation onto related concepts which can be utilized to address the problem of sparsity in large overlay models. Gauch et al. [22] deliver a detailed comparison of different variations of the aforementioned user profile representations and discusses several methods for explicit and implicit collection of user information.

- **Knowledge** as a user feature enjoys the most significance in Adaptive Educational systems, often beeing the only feature modelled. The simplest representation of user knowledge is the scalar model which expresses the degree of knowledge in a particular domain (regarded as a whole) on a predefined scale of either quantitative (e.g. from 0 to 5) or qualitative (e.g. excellent, good, average, etc.) kind. The more advanced structural model, in contrast, divides the body of domain knowledge into fragments (e.g. indicated by ontology concepts) and estimates user's knowledge level for each fragment. An example of this model implemented in the Human Resource domain is presented in [59].

- **Background** of a user relates to a collection of features regarding previous experience outside the core domain of a particular system and may include, for example, profession, certain role within a corporation, work experience in related areas, demographics, language, etc. As argued in [12], most systems do not require detailed information about user background, therefore the common way to model user background is a simple stereotype model.

- **Goals and Tasks** represent the user's immediate purpose for the interaction with an adaptive system. Especially in search scenarios, goals and tasks may also be viewed as context of a given query, which has a great impact on the quality of results delivered by (recommender) systems. For example, a user might be buying items for his or her personal use (1), items which are work related (2) or intended as a gift (3). In those cases user's personal interests have diminishing impact (from 1 to 3) on the quality of recommendations. Consequently, Anand and Mobasher [2] propose a more complex user model, distinguishing long-term interests from short-term goals, which takes this kind of contextual information better into account. The current goal of a user can be modeled as an overlay of a predefined goal catalogue of independent goals. More advanced approaches utilize a goal/task hierarchy decomposing top-level goals into sub-goals at lover hierarchy levels and/or introduce additional relations between goals/tasks in form of an ontology [33]. Due to the short-lived character of user goals as well as the difficulty and impreciseness of goal recognition most system rely on explicit specification of the current user goal.

- **Context.** In general, context can be described as additional mainly short-term information about the circumstances, objects, or conditions surrounding a user (cf. [47]). In this sense, the border between traditional features of a user model described above and context is not always clearly defined. Furthermore, user and context modeling are interrelated, since many user models incorporate context features and similar techniques are applied for modeling [12]. There also exist integrated frameworks for modeling of both context and user features - for instance the general user model and context ontology GUMO [24] represented in OWL. In particular, most commonly used categories of contextual information refer to:

  User platform. Especially since the wide proliferation of various kinds of mobile devices, early context-aware systems were mainly concerned with platform adaptation [7]. Rendering content differently for desktop and mobile devices based on screen size or bandwidth are examples of the application of platform-oriented context.

  Physical context includes such factors as current location and time. User location is usually represented in a coordinate-based or zone-based manner, depending on the location sensing. In context-aware adaptive systems this kind of information is used for finding nearby objects of interest. Time-related factors, such as weekday or opening hours, may be used to impose additional search constrains. The most prominent examples of applications utilizing physical context can be found in the domains of tourism and visitor guides [6] as well as cultural heritage and museum guides [65].

  Human Dimension includes personal and social context. Example features of personal user context are health, mood, affective state, etc., which determination, however, greatly depends on the appropriate sensory input or explicit specification by the user. Social user context may be represented, for example, by people accompanying the user during interaction with the adaptive system (e.g. while looking for a restaurant for a group of people) or by the user's social network. Especially the latter has increasingly been analyzed within the research community exploring social graphs for improved recommendations [35].

4.3.3 Ontological Model for Representing Context Information

To capture the various different context dimensions, including the context of user models, we propose a modular ontological representation model as shown in Figure 4.4.

The top-level ontologies represent general concepts relating to temporal, spatio, event, situation and process context. These concepts can be specialized by domain specific ontologies e.g. for particular user models or information models. The task ontologies relate the context model to the behavioral models such as business process models, semiotic collaboration structures, information search and discovery activities etc. The application specific ontology models

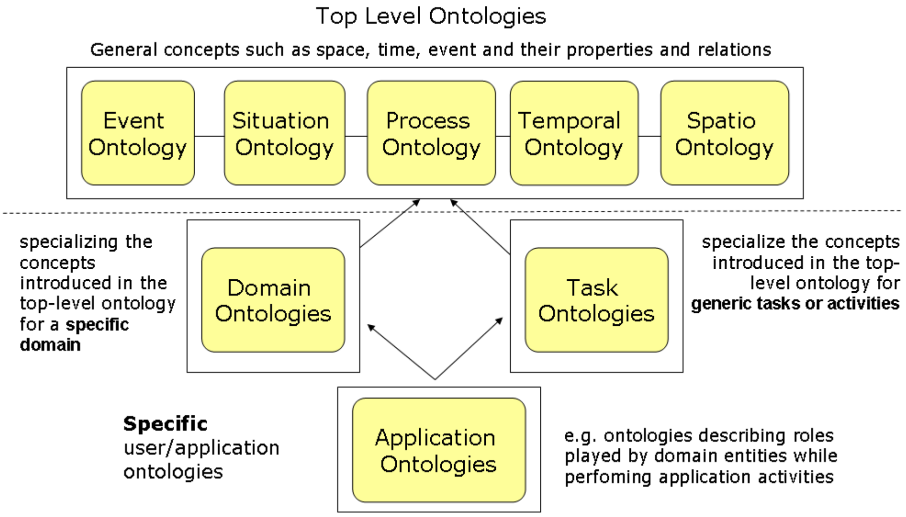

Top Level Ontologies
General concepts such as space, time, event and their properties and relations
Event Ontology
Situation Ontology
Process Ontology
Temporal Ontology
Spatio Ontology
specializing the concepts introduced in the top-level ontology for a specific domain
Domain Ontologies
Task Ontologies
specialize the concepts introduced in the top-level ontology for generic tasks or activities
Specific user/application ontologies
Application Ontologies
e.g. ontologies describing roles played by domain entities while perfoming application activities

Figure 4.4: Modular Ontology Model for Representing Context Information

introduce specific terminologies often on the IT application level, e.g. involved services, application activities, involved entities, data models etc.

Figure 4.5 exemplarily shows the top-level situation ontology. Detecting relevant situations is important for intelligent information dissemination strategies addressing the problem of "*the right information, to the right person, in the right situation*".

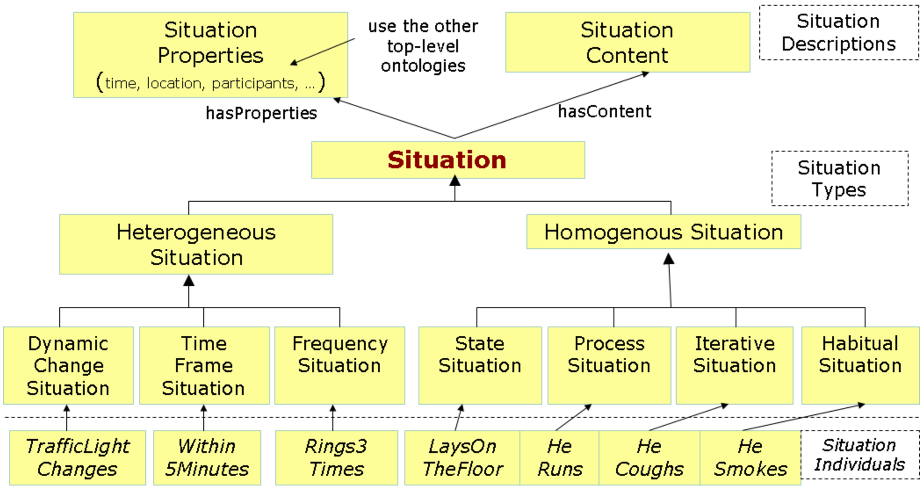


Figure 4.5: Example - Situation Ontology

This model distinguishes between situation types, heterogenous situations, which consist of different heterogenous events and situations, and homogenous situations, which consists of one homogeneous situation type. The two situation categories are distinguished in more specialized situation types. A situation consists of its properties such as time, location, participants (e.g. events initiating and terminating the situation) and the situation content which contains the situation data. The properties of situations relate to the other top-level ontologies and their specializations in particular domain and task ontologies.

These ontologies can be used to semantically model context information and user context and used as background knowledge in semantic search for personalization and contextualization of the different aspects of search as listed in Figure 4.3 and described in the previous subsection 4.3.2.

4.3.4 Conclusion and Outlook

In this section, we focused on various aspects of personalized and context-aware access to large amounts of heterogenous Web resources. First, we concentrated on the user model which is the central component of every adaptive or adaptable system. We provided an overview of several kinds of user-related information with a strong emphasis on user context, and discussed various aspects of modeling. We then depicted a general and highly extensible modular ontological model for representing context information. We argued that Semantic Web technologies not only provide semantic search and recommender systems with a more precise understanding of the application domain, formalized in ontologies, but can also be used for a richer representation of user related and context information.

Since the tasks of personalization and contextualization are highly interrelated, our future research on contextualization of search will build on methods and tools developed in the first stage of the CSW project. In close cooperation with our industrial partners, we will be pursuing further development of approaches for semantic web recommender systems, extending them with context-aware features.

Chapter 5

Conclusion and Outlook

Based on the state of art and requirements analysis described in the last report [43], this report addresses conceptual solutions and first prototypical implementations which further advance CSW application domains such as multimedia content, distributed systems and knowledge, and pragmatic context.

In close collaboration with the project's industry partners, we will now work on the prototypical implementation and evaluation of our proposed new CSW solution approaches in the next milestone of the project. The applied research methodologies will build on the results achieved in the completed working packages of the first phase of the project - see [16, 42, 41].

Appendix A

Work Packages

Work package 3	**Searching non-textual data (multimedia search)**	02/11-01/13
WP 3 Task 3.2	Conception of a method for knowledge retrieval from non-textual corporate data	05/11-07/11
WP 3 Task 3.3	Conceptual and prototypical implementation of a semantic search system over multimedia data based on the results of WP1	08/11-02/12

Work package 4	**Search contextualization**	02/11-01/13
WP 4 Task 4.2	Conception of a method for modeling user (co-worker) context and contextualization of search results	05/11-07/11
WP 4 Task 4.3	Conceptual and prototypical implementation of contextual search based on results from WP 2	08/11-02/12

Work package 7	**Dynamic access to distributed knowledge**	02/11-01/13
WP 7 Task 7.2	Conception of a method for (i) integrating knowledge from distributed heterogeneous sources and (ii) derivation of new knowledge, including identification of trends, corporate structures, or potential problems	05/11-07/11
WP 7 Task 7.3	Partial prototypical implementation	08/11-02/12

Work package 8	**Ontology and knowledge evolution through collaborative work**	02/11-01/13
WP 8 Task 8.1	State-of-the-art survey on ontology and knowledge evolution; adaption of ontology and knowledge evolution principles and methods for the application in the corporate context	02/11-04/11

WP 8 Task 8.2	Design of a semantic method for the semi-automated evolution of ontologies or knowledge bases by analysing collaborative work	05/11-07/11
Work package 11	**Ontology cost models for enterprises**	02/11-01/13
WP 11 Task 11.3	Prototypical implementation of a tool for intra-corporate ontology development cost estimation	08/11-01/12
Work package 12	**Ontology evaluation**	02/11-01/13
WP 12 Task 12.2	Conception of a method for ontology evaluation with regard to usage criteria relevant for enterprises, reusability, and adaptation	05/11-07/11
WP 12 Task 12.3	Conceptual and prototypical implementation of a human-centric ontology evaluation framework	08/11-02/12
Work package 14	**Personalization and Context in the Corporate Semantic Web**	02/11-01/13
WP 14 Task 14.1	Design of personalized search on heterogenous data	02/11-02/12
WP 14 Task 14.2	Design of ontological representations for context information	12/11-02/12

Appendix B

Acknowledgement

This work has been partially supported by the "InnoProfile-Corporate Semantic Web" project funded by the German Federal Ministry of Education and Research (BMBF).

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749, 2005.

[2] Sarabjot Singh Anand and Bamshad Mobasher. From web to social web: Discovering and deploying user and content profiles. chapter Contextual Recommendation, pages 142–160. Springer-Verlag, Berlin, Heidelberg, 2007.

[3] Lora Aroyo, Natalia Stash, Yiwen Wang, Peter Gorgels, and Lloyd Rutledge. Chip demonstrator: Semantics-driven recommendations and museum tour generation. In Jennifer Golbeck and Peter Mika, editors, *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[4] Sören Auer and Heinrich Herre. RapidOWL — An Agile Knowledge Engineering Methodology. In Irina Virbitskaite and Andrei Voronkov, editors, *Perspectives of Systems Informatics*, volume 4378, pages 424–430. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[5] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development, 2001.

[6] Christian Becker and Christian Bizer. Exploring the geospatial semantic web with dbpedia mobile. *Web Semant.*, 7:278–286, December 2009.

[7] Silvia Berti, Giulio Mori, Fabio Paternï, and Carmen Santoro. An environment for designing and developing multi-platform interactive applications. In *Proceedings of HCITALY2003*, pages 7–16. University of Turin, 2003.

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

[9] Yolanda Blanco-Fernández, José J. Pazos-Arias, Alberto Gil-Solla, Manuel Ramos-Cabrer, Martín López-Nores, Jorge García-Duque, Ana Fernández-Vilas, Rebeca P. Díaz-Redondo, and Jesús Bermejo-Muñoz. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Know.-Based Syst.*, 21:305–320, May 2008.

[10] Barry Boehm. *Software Engineering Economics.* Prentice-Hall, Englewood Cliffs, N.J, 1981.

[11] Barry W. Boehm, Bradford Clark, Ellis Horowitz, J. Christopher Westland, Raymond J. Madachy, and Richard W. Selby. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. *Annals of Software Engineering*, 1:57–94, 1995.

[12] Peter Brusilovsky and Eva Millán. The adaptive web. chapter User models for adaptive hypermedia and adaptive educational systems, pages 3–53. Springer-Verlag, Berlin, Heidelberg, 2007.

[13] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: implementing the semantic web recommendations. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 74–83, New York, NY, USA, 2004. ACM.

[14] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[15] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models.* Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1986.

[16] Gökhan Coşkun, Ralf Heese, Markus Luczak-Rösch, Radoslaw Oldakowski, Ralph Schäfermeier, and Olga Streibel. Towards corporate semantic web: Requirements and use cases. Technical Report B 08-09, Freie Universität Berlin, August 2008.

[17] Valerie Cross and Xueheng Hu. Using semantic similarity in ontology alignment. In Pavel Shvaiko, Jïrïme Euzenat, Tom Heath, Christoph Quix, Ming Mao, and Isabel F. Cruz, editors, *OM*, volume 814 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[18] Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, Marta Sabou, and Enrico Motta. Characterizing knowledge on the semantic web with watson. In Raul Garcia-Castro, Denny Vrandecic, Asuncin Gmez-Prez, York Sure, and Zhisheng Huang, editors, *EON*, volume 329 of *CEUR Workshop Proceedings*, pages 1–10. CEUR-WS.org, 2007.

[19] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and Y. Kompatsiaris. A survey of semantic image and video annotation tools. *Knowledge-driven multimedia information extraction and ontology evolution*, pages 196–239, 2011.

[20] L. Ding and Tim Finin. Characterizing the semantic web on the web. In *Proceedings of the 5th International Semantic Web Conference*, 2006.

[21] Paul Doran, Valentina Tamma, and Luigi Iannone. Ontology module extraction for ontology reuse: an ontology engineering perspective. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 61–70, New York, NY, USA, 2007. ACM.

[22] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User profiles for personalized information access. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 2, pages 54–89. 2007.

[23] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research (JAIR)*, 31:273–318, 2008.

[24] Dominik Heckmann, Eric Schwarzkopf, Junichiro Mori, Dietmar Dengler, and Alexander Kr̈ıner. The user model and context ontology gumo revisited for future web 2.0 extensions. In Paolo Bouquet, J̈ır̈ıme Euzenat, Chiara Ghidini, Deborah L. McGuinness, Luciano Serafini, Pavel Shvaiko, and Holger Wache, editors, *C&O:RR*, volume 298 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[25] Ralf Heese, Markus Luczak-Rösch, Adrian Paschke, Radoslaw Oldakowski, and Olga Streibel. One click annotation. In *SFSW*, 2010.

[26] Martin Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, EKAW '08, pages 329–346, Berlin, Heidelberg, 2008. Springer-Verlag.

[27] M. Horvat, S. Popovic, N. Bogunovic, and K. Cosic. Tagging multimedia stimuli with ontologies. *Arxiv preprint arXiv:0903.0829*, 2009.

[28] Jeff Howe. The rise of crowdsourcing. *Wired*, 14(6), 2006.

[29] David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Denny Vrandečić, and Aldo Gangemi. Unit Tests for Ontologies. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278, pages 1012–1020. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[30] Chris F. Kemerer. An Empirical Validation o Estimation Models. *Communications of the ACM ACM*, 30(5):416–429, May 1987.

[31] W. Lee, T. Bürger, F. Sasaki, V. Malaisé, F. Stegmaier, and J. Söderberg. Ontology for media resource 1.0. *W3C Working Draft*, 18, 2009.

[32] Roza Lémdani, Géraldine Polaillon, Nacéra Bennacer, and Yolaine Bourda. A semantic similarity measure for recommender systems. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 183–186, New York, NY, USA, 2011. ACM.

[33] Yun Lin and Arne Sølvberg. Goal annotation of process models for semantic enrichment of process knowledge. In *Proceedings of the 19th international conference on Advanced information systems engineering*, CAiSE'07, pages 355–369, Berlin, Heidelberg, 2007. Springer-Verlag.

[34] Markus Luczak-Röch and Markus Bischoff. Statistical analysis of web of data usage. In *Joint Workshop on Knowledge Evolution and Ontology Dynamics (EvoDyn2011)*, 2011.

[35] Ashwin Machanavajjhala, Aleksandra Korolova, and Atish Das Sarma. Personalized social recommendations - accurate or private? *PVLDB*, 4(7):440–450, 2011.

[36] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 117–, Washington, DC, USA, 2002. IEEE Computer Society.

[37] S. Middleton, H. Alani, N. Shadbolt, and D. De Roure. Exploiting Synergy Between Ontologies and Recommender Systems. In *11th International World Wide Web Conference, Semantic Web Workshop*, pages 41–50, 2002.

[38] Peter Brusilovsky; Alfred Kobsa; Wolfgang Nejdl, editor. *The Adaptive Web: Methods and Strategies of Web Personalization.* Springer Verlag, 2007.

[39] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[40] Ilia Ovsiannikov and Michael Arbib. *Annotator: Annotation Technology for the WWW*, chapter 5.3, pages 255–264. Academic Press, Inc, 2001.

[41] Adrian Paschke, Gökhan Coşkun, Dennis Hartrampf, Ralf Heese, Markus Luczak-Rösch, Mario Rothe, Radoslaw Oldakowski, and Ralph Schäfermeierand Olga Streibel. Realizing the corporate semantic web: Prototypical implementations. Technical Report TR-B-10-05, Freie Universität Berlin, 2010.

[42] Adrian Paschke, Gökhan Coşkun, Ralf Heese, Markus Luczak-Rösch, Radoslaw Oldakowski, Ralph Schäfermeier, and Olga Streibel. Realizing the corporate semantic web: Concept papers. Technical Report TR-B-08-09, Freie Universität Berlin, April 2009.

[43] Adrian Paschke, Gökhan Coskun, Ralf Heese, Radoslaw Oldakowski, Mario Rothe, Ralph Schäfermeier, Olga Streibel, Kia Teymourian, and Alexandru Todor. Corporate Semantic Web - Report IV: State of the Art Analysis. Technical Report TR-B-11-07, Freie Universität Berlin, 2011.

[44] E. Paslaru Bontas Simperl, C. Tempich, and Y. Sure. ONTOCOM: A Cost Estimation Model for Ontology Engineering. *The Semantic Web-ISWC 2006*, pages 625–639, 2006.

[45] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.

[46] N. Bennacer R. Lïmdani, G. Polaillon and Y. Bourda. A semantic similarity measure for recommender systems. Technical report, SUPELEC Systems Sciences (E3S), 2011.

[47] Anand Ranganathan and Roy H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Comput.*, 7:353–364, December 2003.

[48] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Arxiv preprint cond-mat/0603718*, 2006.

[49] L.A. Rowe, J.S. Boreczky, and C.A. Eads. *Indexes for user access to large video databases.* Citeseer, 1994.

[50] Ahu Sieg, Bamshad Mobasher, and Robin Burke. Web search personalization with ontological user profiles. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 525–534, New York, NY, USA, 2007. ACM.

[51] Elena Simperl, Igor O. Popov, and Tobias Bürger. ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554, pages 248–262. Springer, Berlin, Heidelberg, 2009.

[52] R. Sorschag. A flexible object-of-interest annotation framework for online video portals. *Future Internet*, 4(1):179–215, 2012.

[53] Olga Streibel. Semantic-based learning method for trend recognition in simple hybrid information systems. In *online proceedings of Doctoral Consortium at Conference on Advanced Information Systems Engineering CAISE2008*, 2008.

[54] Olga Streibel. Trend mining with semantic-based learning. In *online proceedings of Ph.D.Symposium at European Semantic Web Conference ESWC2008*, pages 71–72, 2008.

[55] Olga Streibel. Mining trends in texts on the web. In *Proceedings of the DC at Future Internet Symposium 2010*, pages 80–90. CEUR. Vol-623, 2010.

[56] Olga Streibel and Rehab Alnemr. Trend-based and reputation-versed personalized news network. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents, SMUC2011*, pages 3–10. ACM, 2011.

[57] Olga Streibel and Malgorzata Mochol. Trend ontology for knowledge-based trend mining on textual information. In *IEEE Computer Society Proceedings of 7th International Conference on Information Technology : New Generations, ITNG2010, April 2010*, pages 1285–1288. IEEE Computer Society, 2010.

[58] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Boris Villazón-Terrazas. How to Write and Use the Ontology Requirements Specification Document. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II*, OTM '09, pages 966–982, Berlin, Heidelberg, 2009. Springer-Verlag.

[59] Robert Tolksdorf, Malgorzata Mochol, Ralf Heese, Rainer Eckstein, Radoslaw Oldakowski, and Christian Bizer. Semantic-web-technologien im arbeitsvermittlungsprozess. *Wirtschaftsinformatik*, 1, 2006.

[60] D. Van Deursen, R. Troncy, E. Mannens, S. Pfeiffer, Y. Lafon, and R. Van de Walle. Implementing the media fragments uri specification. In *Proceedings of the 19th international conference on World wide web*, pages 1361–1364. ACM, 2010.

[61] R.Q. Wang and F.S. y Kong. Semantic-enhanced personalized recommender system. In *Proc. of the Int. Conference on Machine Learning and Cybernetics*, volume 7, pages 4069–4074, 2007.

[62] Dave West, Jeffrey S. Hammond, Mike Gilpin, and David D'Silva. The Forrester Wave: Agile Development Management Tools, Q2 2010. Technical report, Forrester Research, Inc., 2010.

[63] William E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.

[64] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. In *Proceedings of the 10th International Conference on Conceptual Structures (ICCS)*, pages 92–196, London, UK, 2002. Springer-Verlag.

[65] Andreas Zimmermann, Marcus Specht, and Andreas Lorenz. Personalization and context management. *User Modeling and User-Adapted Interaction*, 15(3-4):275–302, August 2005.