Corporate Semantic Web
Report VI

Validation and Evaluation

Technical Report TR-B-13-01

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

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

31 January 2013

INNOPROFILE
UNTERNEHMEN REGION
Die BMBF-Innovationsinitiative
Neue Länder

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Corporate Semantic Web
Report VI
Validation and Evaluation

Adrian Paschke Gökhan Coskun Marko Harasic
Ralf Heese Radoslaw Oldakowski Ralph Schäfermeier
Olga Streibel Kia Teymourian Alexandru Todor

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

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

Technical Report TR-B-13-01

31 January 2013

Abstract

In this technical report, we present 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, developed according to the working plan outlined in Technical Report TR-B-12-04 [34].

The prototypes complete the next milestone on the path to an integral Corporate Semantic Web architecture based on the three pillars Corporate Ontology Engineering, Corporate Semantic Collaboration, and Corporate Semantic Search, as envisioned in [33].

Contents

Chapter 1

Introduction

In this technical report, we present prototypical implementations and evaluations 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, developed according to the working plan outlined in Technical Report TR-B-12-04 [34].

The proof-of-concept prototypes complete the next milestone on the path to an integral Corporate Semantic Web architecture based on the three pillars Corporate Ontology Engineering, Corporate Semantic Collaboration, and Corporate Semantic Search, as envisioned in [33]. The prototypes were partially developed in tight co-operation with our industrial partners and evaluated on the basis of industrial use cases and demonstrators.

In chapter 2 we present our results in ontology evaluation and cost estimation in agile ontology engineering processes. This addresses the pragmatic aspects of (re-)using and engineering ontologies in enterprise settings.

In chapter 3 we present the evaluation and experimental results of the developed concepts for Corporate Semantic Collaboration.

Chapter 4 covers the implementation and evaluation results in the Corporate Semantic Search pillar with a specific focus on search in multi-media data, semantic context-based recommendations, and integration of personalized views.

Chapter 2

Corporate Ontology Engineering

In traditional development methods of knowledge-based systems creating ontologies are heavyweight processes, including detailed domain and application analysis. Before the ontology is deployed it passes through different tests until it reaches a satisfying maturity level. Maintenance in this case might be a rather small issue, as it is very unlikely that further refinement is necessary while the ontology is in use, apart from small corrections.

But bringing ontologies into enterprise environments poses new challenges for ontology engineering methodologies. Running businesses have strict constraints with respect to the capital expenditure as well as the operational expenditure. Long-lasting and cumbersome development processes with a long-term investment return are not acceptable. In fact, agile processes allowing for quickly exploitable initial versions are desired.

In the context of agile ontology engineering methodologies cost-estimation techniques as well as efficient reuse and maintenance support needs special attention. On the one hand reusing existing ontologies reduces investment costs. On the other hand ontology maintenance in case of agile processes can be considered as equal to forward engineering forming the overall evolution process. In this regard maintenance does not include only low-level activities like adding new elements, updating, refining, merging, and removing existing elements it also comprises the complete refactoring. Thus, for the sake of efficiency it is essential to understand and to quantify the overall improvement in order to justify the maintenance process. That means it is important to estimate the cost for the maintenance on the one hand and the benefit and profit on the other hand.

An important assumption for efficient reuse and maintenance is the ability to evaluate the ontology adequately. Because the decision whether to reuse an ontology depends on the degree of its reusability for the envisioned system and on the necessary customization. With respect to maintenance it is important that the overall quality is increased by each step.

In this regard this chapter presents ontology evaluation in section 2.1 and cost estimation considerations and factors in agile ontology engineering processes in section 2.2.

2.1 Ontology Evaluation (WP 12)

Development strategy in IT seeks for abstraction, encapsulation and reusability in various levels. This caused different paradigms like object-oriented programming, agent-oriented programming, aspect oriented programming and different techniques like middleware and application containers. The distinction between the program logic and the information model is suggested throughout these approaches. The reason for this is twofold, avoiding dependency between the model and the programming logic, and secondly, allowing for reusable components. For that reason, reusability is an inherent feature of ontologies, which are the semantically enriched information models of knowledge-based systems. According to (Dzbor & Motta 2008) "the reuse of existing, possibly imperfect, ontologies becomes the key engineering task."

The reuse process commonly starts with the intention to utilize ontologies in an envisioned IT system. That means that the developer has an application and a domain in mind. Based on this the developer starts searching for candidate ontologies, which might be reused. Different search engines (dïAquin, Sabou, et al. 2007) and ontology libraries (dïAquin & Natalya F Noy 2012) are available to support this discovery process. Having obtained a list of potential candidate ontologies an analysis and decision taking step has to be done. The coverage and level of detail of each candidate has to be evaluated, in order to answer the question if an ontology is reusable for the targeted system. If it is, the second question is, to which extend it can be reused and whether it needs some kind of customization. Reuse can range from an inspiring input up to the complete adoption without any customization. It is also possible that a candidate is reused partially, which would assume some modularization step to be taken. It is very important that these decisions are taken quickly and correctly. If the analysis process cannot be done efficiently or the decision is made wrong, the reuse effort would lead to waste of time and resource, although its primary motivation was to save resources and time.

Careful documentation of the development process and the created artifact is broadly accepted as an important means to support reuse. It is frequently used in the field of Software Engineering, where tools like javadoc are very popular. In the field of ontology engineering in contrary the lack of good documentation makes reuse 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.

2.1.1 Understanding Ontologies

In previous work [34] we propose a structure-based ontology partitioning technique to create concept groups for the documentation in a (semi-) automatic way. This technique was evaluated by comparing the results with existing concept groups from documentations. In this regard the existing groups have been considered as being a gold standard because they were created by the ontology developers. An additional assumption was that this kind of grouping within a documentation is a good support to understand the content of an ontology. This in fact was to be proved.

We executed a user study where we created two groups of users. The groups had to answer questions about an ontology where the first group had access only to an alphabetically sorted list of concepts while the second group of users had to answer the same questions but had access to concept groupings. These groupings were created with our structure-based partitioned approach. If the aforementioned assumption is true, the second group of users will understand the ontology faster and will answer the questions quickly. A questionnaire is a well-known technique to measure the knowledge of a user.

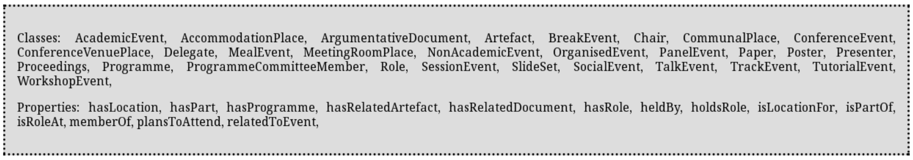
The Setup

For the user study it was important that the users have basic experience with ontologies and are familiar with concepts of semantic technologies. We selected the following three ontologies which model a very generic domain:

1. **ECOS**: Enterprise Competence Organization Schema
2. **PO**: BBC Programmes Ontology
3. **SWCO**: Semantic Web Conference Ontology

It is important not to choose an ontology which describes a very technical domain demanding for specific knowledge, so all users have the same preconditions. Each user got either a list of concepts as shown in Figure 2.1 or concept groups as shown in Figure 2.2 and had to answer the following questions:

- **ECOS**: Is it possible to describe finished projects of companies? (expected answer: yes)
- **PO**: Is it possible to describe the actors taking place within a show? (expected answer: yes)
- **SWCO**: Is it possible to describe points of interest in city where conferences are taking place? (expected answer: no)



Classes: AcademicEvent, AccommodationPlace, ArgumentativeDocument, Artefact, BreakEvent, Chair, CommunalPlace, ConferenceEvent, ConferenceVenuePlace, Delegate, MealEvent, MeetingRoomPlace, NonAcademicEvent, OrganisedEvent, PanelEvent, Paper, Poster, Presenter, Proceedings, Programme, ProgrammeCommitteeMember, Role, SessionEvent, SlideSet, SocialEvent, TalkEvent, TrackEvent, TutorialEvent, WorkshopEvent,

Properties: hasLocation, hasPart, hasProgramme, hasRelatedArtefact, hasRelatedDocument, hasRole, heldBy, holdsRole, isLocationFor, isPartOf, isRoleAt, memberOf, plansToAttend, relatedToEvent,

Figure 2.1: Concepts and properties of SWCO as a HTML list

To prove how a grouping of concept supports the decision about the reusability we measured the time that was needed to answer the above questions. If the user who got the concept grouping were able to answer faster than the users who got just a list of concepts the benefit of such a grouping would be shown.

Results

Table 2.1 shows the results of the experiments. Each row represents the results for each ontology. The table contains three different part. The first part shows the results for the experiments with a concept list. The second part shows

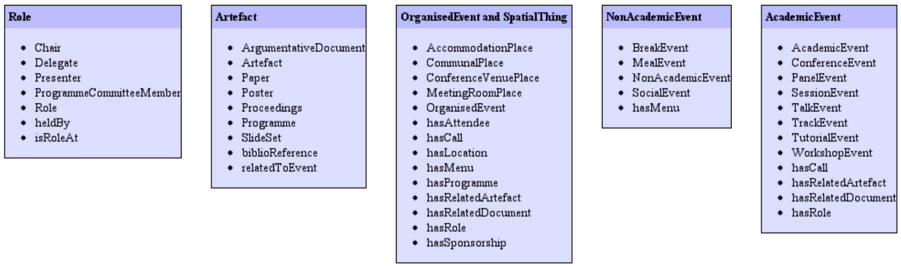


Figure 2.2: Concept grouping of the SWCO created automatically with WTC algorithm

Table 2.1: Results of the experiment (time in seconds)

	List			Grouping 1 (FGC)			Grouping 2 (WTC)		
	Exp. 1	**Exp. 2**	**AVG**	**Exp. 3**	**Exp. 4**	**AVG**	**Exp. 5**	**Exp. 6**	**AVG**
ecos	34	40	**37**	11	20	**15.5**	35	54	**44.5**
po	15	54	**34.5**	16	9	**12.5**	13	3	**8**
swco	37	23	**30**	36	16	**37**	35	28	**31.5**

the results for the experiments with the concept grouping which was created automatically the Fast Greedy Community algorithm. Finally, the third part shows the results for the experiments with the concept grouping created by the Walktrap Community algorithm. Each part contains an additional row which shows the average value.

Figure 2.3 visualizes the average values. As this experiment is done only with 6 persons the results are not significant. However, it is possible to identify a trend.

Conclusion

In case of ECOS the answer time for the concept listing and the concept grouping with WTC are about the same value (34-54 seconds), while the answer time in case of concept grouping with FGC is significantly shorter (11sec and 20sec). After interviewing the user about their search process the reason for this could be understood. The naming of the grouping in case of FGC was better and helped the users to focus on the relevant group while the naming of the concept grouping in case of WTC caused the user to start searching in wrong groups.

In case of PO the answer time of the experiments with the concept list is not very different compared to the answer time of the experiments with the concept grouping. This reason for this is that the important concept “actor” is in an alphabetically list at the beginning. An alphabetically sorted list can speed up the decision if the search concepts are represented by exactly the same word and not by a synonym for example and the list is short enough.

The answer time for the third question is in each case about the same high value. This is caused by the negative answer, which requires a search about the whole ontology. In contrary, a positive answer can be given as soon as the necessary concepts have been found without the need to perform a complete search. In three of four cases the positive answer was given faster in case of an available concept grouping. This indicates that the decision time and therefore the efficiency in reusing depends on a search process for important concepts. If

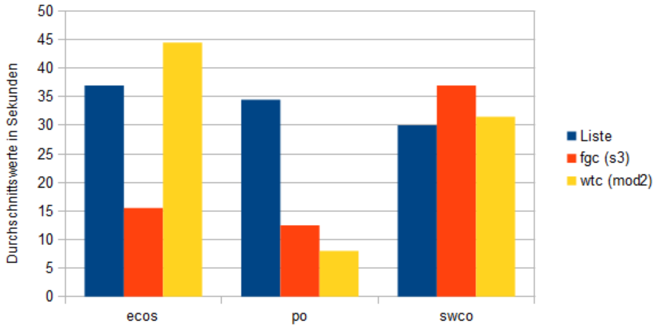


Figure 2.3: Average values of the three ontology ECOS, PO, and SWCO

the user is able to focus on the relevant parts which is only possible with expressive group names the user will be much faster to decide about the reusability of and ontology.

2.2 Cost Factors in Agile Ontology Development (WP 11)

In our last technical report we proposed a novel approach for supporting developers engaged in agile ontology development in estimating project effort by combining static cost models with project effort assessment practices from the agile software development domain [34].

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 [2].

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 [2].

In this working package, we combine *ONTOCOM* [35] 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. *ONTOCOM* is an algorithmic cost model derived from the software cost estimation model *COCOMO* [3, 4]. 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 [40]. 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.

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.

The remainder of this section is organized as follows: First, we present the results of two experiments we ran in order to validate our approach. Then, we present a survey we conducted among our industrial partners. We conclude by discussing the results and giving a prospect on future work.

2.2.1 Experiments

We conducted two experiments with ten and six teams, respectively.

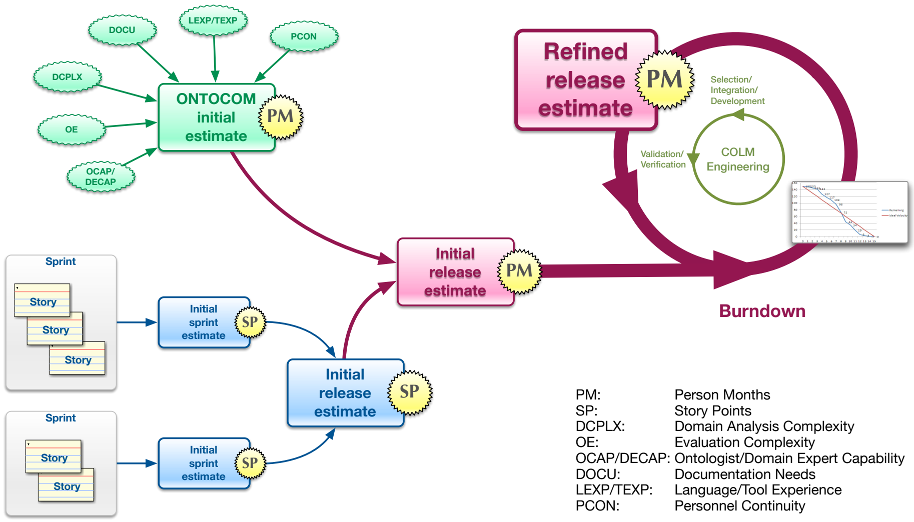


Figure 2.4: Hybrid cost model for agile ontology development processes

Setup

The setup of the experiment was as follows: Each team was presented with a scenario and asked to systematically develop an ontology for the given scenario, making use of agile development principles.

The teams were asked to plan their projects according to the following phases:

- **Requirements elicitation**: Gather requirements in the form of competency questions which the intended application should be able to answer on the basis of the ontology (as proposed in [44]).

- **Initial cost estimation:** Once a sufficient number of competency questions have been gathered in order to describe the expected aspects of the model, estimate the effort using the ONTOCOM model and an independent estimate.

  - **Ontology size:** Estimate the prospective size of the ontology in terms of numbers of axioms.
  - **Cost drivers:** Asses the relevance of each of the cost drivers used by the ONTOCOM model in the context of the project.
  - **Independent estimate:** Make an estimate for each of the competency questions in terms of effort or story points, based on the project team's own experience.

- **Sprints:** Once the requirements have been defined and an initial effort estimation has been made, start modeling. Split up the work along the given tasks and competency questions, and perform the work by completing several sprints.

In addition to the advices on how to proceed, the following tasks had to be accomplished:

- Model the ontology according to the competency questions.
- Localize the ontology, for example by adding labels in different languages.
- Search for existing ontologies that cover the domain and goals of the applications scenario and integrate them.

Results

The outcome of the experiments consisted of ten ontologies and project documentation containing

- stories in the form of competency questions
- effective effort spent on each development task
- documentation of factors influencing the effort spent

The size of the resulting ontologies varied between 45 and 168 entities. The effort spent for the entire development process, including the requirements elicitation phase, the sprints, ontology integration tasks, ontology localization tasks, and team discussions varied between one and three entire working days, which coincided well with the predictions made by the developers but deviated significantly from the predictions obtained by using ONTOCOM. However, most teams faced difficulties with implications for the development effort.

The problems reported by the participants fell into either of the following classes:

- Tool support **(TOOL)**
- Overlapping roles/requirements **(OLREQ)**
- Unclear requirements **(UCREQ)**
- Necessity to refactor during a later sprint **(REF)**
- Initial difficulties determining the best way of modeling certain facts **(MOD)**

The identified classes of causes for project backlog can be further broken down into the following categories:

TOOL: Causes for project backlog related to tool support vary from general problems operating the modeling tool (Usability issues, modeling tool or reasoner crashing due to uncaught errors or too large ontologies).

OLREQ: Participants reported different problems concerning the requirements elicitation process or the following implementation of the requirements.

Common among the problems reported when dealing with requirements was the phenomenon of overlapping roles or requirements. Developers either had difficulties separating concerns of certain concepts at the modeling stage when the concepts in question where involved in multiple (functional and/or non-functional) requirements (e. g. a person in the role as the originator of pieces of information and, at the same time, in the role as a customer) or later refactoring of concepts with ambiguous roles.

Along these lines, participants reported problems deciding whether concepts should be modelled as concepts (classes), individuals (instances) or relationships role ambiguity was present, resulting in later refactoring or redundant modeling of the same concepts.

UCREQ: Another common difficulty encountered was missing specificity of requirements due to underspecification during first customer communication. Participants also reported difficulties in determining the exact boundaries of the domain of interest, leading to disagreement on where to stop modeling and extended need for communication.

REF: Participants reported need for late refactoring for two reasons, the first being inconsistencies introduced early during the requirements specification phase. These differ from the problem of unclear requirements because this problem category cannot be attributed to the nature of the requirements alone but to discrepancies between the formalisms used for representing the requirements and their incompatibility to the modeling language or approach used during the development phase. The second instance of late refactoring was at the stage of relationship introduction. This represents a special case of the first one but is mentioned here because participants reported this case repeatedly.

MOD: Cases where participating project teams encountered project backlog due to modeling problems can be characterized by initial disagreement on the structure of the ontology or the way of modeling complex facts. These lead to further effort spent on communication and planning.

Most of the incidents reported that fall under this category, however, concerned the task of integrating external ontologies. Problems encountered included difficulties in identifying suitable existing ontologies for a given set of requirements, high effort spent on the integration process due to insufficient modularity of the external ontologies, and arising need for late refactoring due to incompatibilities between own and external ontologies.

The frequency of incidents of each of the categories is shown in figure 2.5.

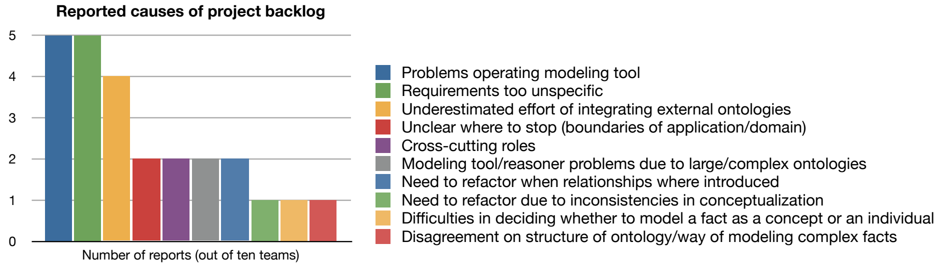


Figure 2.5: Number of reported causes of project backlog out of ten teams.

Cause	Group										**MIN**	**MAX**	**AVG**
	1	2	3	4	5	6	7	8	9	10			
UCREQ					50	50		100	100		**50**	**100**	**75**
OLREQ	6				49	67		50			**6**	**67**	**43**
MOD					6				80		**6**	**80**	**43**
REF					33	50					**33**	**50**	**42**
TOOL	47		22		6	17	33		13	100	**6**	**100**	**34**

Table 2.2: Project backlog by cause and project team, measured in per cent of the overall effort, ordered by average backlog.

While the most backlog in terms of person hours is generated by issues related to unclear requirements, this class of problems can be considered standard in agile development processes and attributed to the very nature of agile methodologies. The agile principle encourages underspecification at an early stage, and therefore need for further communication, requirements refinement, and rescheduling throughout the entire development process is common.

Likewise, refactoring is a usual activity in agile processes, and the high effort spent on refactoring activities can be considered normal.

On the other hand, backlog that can be attributed to factors not induced by agile principles sums up to an average of 120 % of the overall project effort. In particular, discerning overlapping requirements (OLREQ) leads to an average overhead of 43 % of the total project effort. Similarly, modeling complex facts and issues faced with the task of integrating external ontologies due to a lack of modularity and the need to overcome incompatibilites account for 42 % of average project backlog. Issues with tool support when dealing with large complex or large ontologies account for another 35 % average backlog.

2.2.2 Survey

We conducted a survey among our industrial partners, asking for concrete project data from ontology engineering projects where agile principles where applied. Out of eight companies requested to participate, three responded. None of the companies had a project running involving the construction of an ontology during the evaluation period of this work package. Therefore, we presented the participants a questionnaire in order to gather data from previous projects.

The survey was constructed as follows:

First, we collected some key details about the company and the project in question. Then, we asked the participants to assess the importance of each of the relevant cost drivers according to the ONTOCOM model. Then, we collected detailed information about the general process and the course of the project during its different stages according to our model. The questions concerned requirements management and customer communication, performance measurement, factors influencing the project schedule, measures taken, and an overall estimation of the economic value of our approach.

Results of the Survey

While all participants agreed that means for cost prediction of ontology projects would be “nice to have”, they acknowledged the fact that, despite decades of research conducted in the field of software project estimation, formal methods existing to this date are still less reliable than project managers’ experience, at least for small to mid-sized projects. However, in projects with mid-sized or large teams, some means of monitoring the progress of individual sub-teams or team members, enabling project management to intervene in case of backlogs, are considered valuable.

2.2.3 Conclusion and Outlook

The results from the experiments described in section 2.2.1 support our hypothesis that agile ontology development circumvents some of the difficulties of predicting ontology development costs, especially the lack of accuracy of arithmetic cost prediction models. However, hybrid approaches for cost estimation in agile projects involve experience, more accurate documentation and rigid project management.

The experiments reveal further need for methodological and technical support during the development process. While recent research and the activities of the Corporate Semantic Web group have ameliorated the situation to some extent, the experiments reveal a lack of general formal and technical support for modularizing and integrating ontologies. One important obstacle is the intermingling of cross-cutting concerns. Therefore, we argue that formalisms for the separation of cross-cutting aspects in ontologies are imperatively needed.

For a deeper understanding of the economical aspects of ontologies, it would be necessary to assess not only the cost but also the benefit in terms of revenue the deployment of ontologies produces. The wide use of ontologies and ontology-based applications has set off in recent years, and data on generated revenue is sparse at the time of this writing but will be available to increasing degrees in the coming years. However, unlike software systems as a whole, the attribution of revenue change to the deployment of ontologies is not trivial. In order to fully understand and quantify the economic implications of the use of ontologies, further research is needed.

Chapter 3

Corporate Semantic Collaboration

One of the most important part of corporate knowledge can be extracted from corporate environment which can be observed in different types and situations like, corporate user activities, organizational memory, internal or external events. In the previous reports on our research, we presented concepts for the utilization of corporate environment knowledge. We focus on approaches for *"Dynamic Access to Distributed Knowledge"*, *"Ontology and Knowledge Evolution through Collaborative Work"* and *"Semantic Complex Event Processing"*.

In this chapter, we present the evaluation and experimental results of the developed concepts for Corporate Semantic Collaboration. Section 3.1.1 contains the description of eXTS- serious game evaluation. Regarding evaluation of trend ontologies, please see our separate technical report: [43]. In Section 3.2 we describe the user studies about manual annotation using light-weight annotation tools and discuss their results. Section 3.3 presents our experimental results on knowledge-based complex event processing and the process of fusion of external knowledge bases with a stream of events.

3.1 Dynamic Access To Distributed Knowledge (AP 7)

3.1.1 Experimental Evaluation of eXTS

eXTS as a Serious Game

In order to evaluate the racing game we start a test run. The technical setup of the evaluation run is:[1]

- The initial set of words consists of all entities generated during the evaluation of the basic eXTS implementation. The set has a size of 1398 words.
- The game is playable for a period of 14 days to have comparable conditions with the first evaluation. The results of the evaluation are:

[1] This section contains contribution written by Denis Hartrampf

1 There were 34 users that played the game.

2 The players created a total of 1824 taggings.

3 The taggings have no relation, because there was no initial possibility of giving one.

In the following we analyse and compare the productivity of both, the eXTS game and the eXTS website. [51] propose a set of metrics for determining the success of a GWAP. Originally designed for GWAPs, these metrics can also be applied to the eXTS website (and to the eXTS game). The first measure is meant for determining the efficiency of a GWAP and is called the throughput. The authors define the throughput as the number of problem instances solved per human-hour. This means input-output mappings generated or, in the case of eXTS, taggings done. According to the authors the ESP game [41] for example has a throughput of 233 labels per human-hour. The higher the throughput the better. The second measure is the average lifetime play (ALP). It is the total amount of time a game is played by each player averaged across all players. The authors use this measure to express the enjoyability of a game. The higher the ALP is, the greater the enjoyability of the game is considered to be. The ESP game has an ALP of 91 minutes. The last measure combines the two previously described ones. The authors call it expected contribution. It is the product of throughput and ALP. It represents the number of problem instances that a user is expected to solve during her lifetime play. Because the problem instances that are solved by the eXTS game and the website are not the same one could argue that they cannot be compared according to throughput, ALP and expected contribution. For the website the problem is to find a tag and a relation for a word whereas for the game the problem is to only find a tag for a word. This

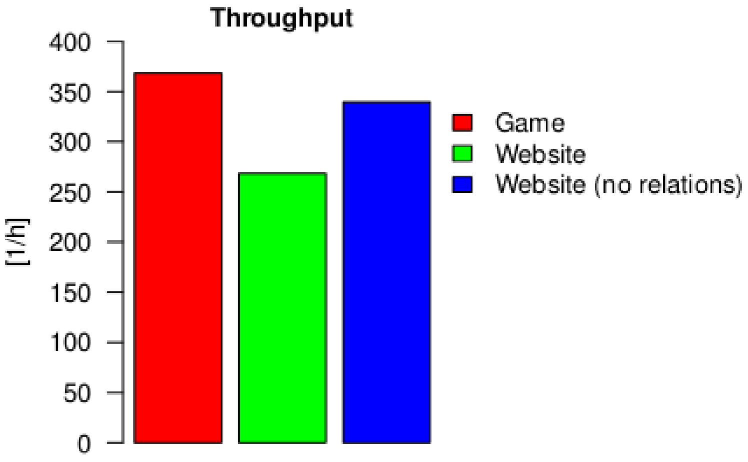


Fig. 5. This bar chart shows the throughput. From left to right for the game, the website and the website without relations.

Figure 3.1: Throughput

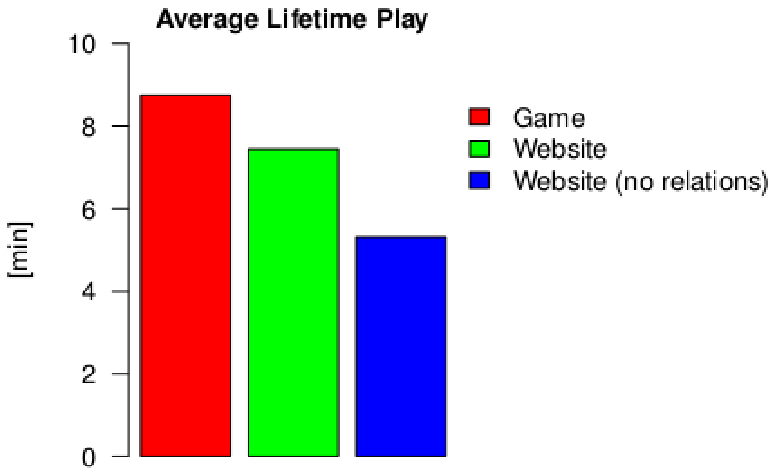


Fig. 4. This bar chart shows the average lifetime play. From left to right for the game, the website and the website without relations.

Figure 3.2: ALP

argument is reasonable, so we want to give a counter-argument. To do this we need some restraints on the data that allow us to consider the problems to be solved in both systems to be equal or at least very alike. To achieve this, only certain data generated by the users of the website are examined. We take only into account assignments from those users, who only committed assignments without relations. This means only users, who did not commit a single assignment with a relation. This reduces the data set, leading to the following facts: There were 30 users using the system. They committed a total of 900 taggings. All of the taggings have no relation. The players of the game had an average lifetime play of 8 minutes and 45 seconds. The users of the website were only active for an average of 7 minutes and 27 seconds and 5 minutes and 19 seconds for the reduced data set.
The eXTS game beats the website in all three measures both with the whole and the reduced data set. But for reasons of fairness it is to say that the ALP for the website as well as the game is quite low compared to e.g. the ESP game, which has an ALP of 91 minutes. So summarizing the observations discussed above it can be said that the game outperforms the website but there is still work to do to improve the fun factor of the game.

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

The Semantic Web envisions a network of semantically-enriched content containing links to explicit, formal semantics. This would then allow to distinguish between the different meanings of a word (e.g., [16]). So far, most of the se-

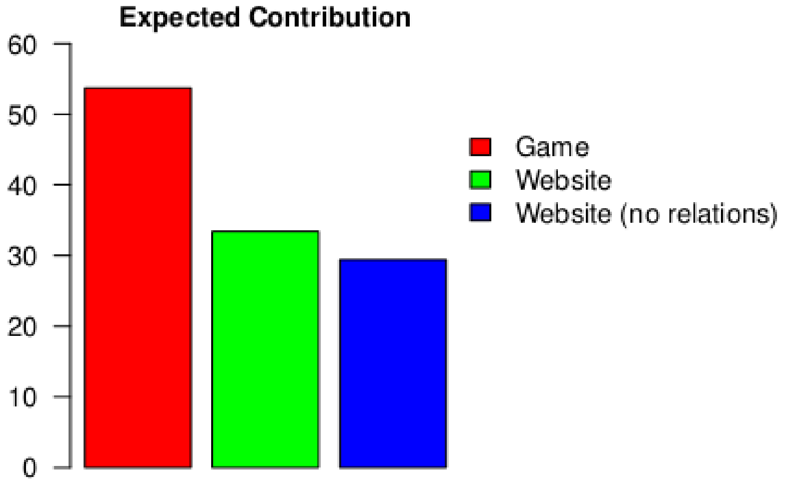


Fig. 6. This bar chart shows the expected contribution. From left to right for the game, the website and the website without relations.

Figure 3.3: Expected Contribution

mantic content available has been generated automatically, typically by using annotation services for existing content. Even though automatic tools are well developed for English language texts [49, 53, 12], other languages are poorly supported. In order to have high quality annotations people may want to follow a semi-automatic approach and revise annotations manually.

The manual annotation step requires tools that hide the complexity of semantic technologies and match the compelling simplicity of Web 2.0 applications: light-weight, easy-to-use, and easy-to-understand. We developed the tool loomp for creating text annotations manually. In a user study based on a paper prototype of loomp participant wanted to create overlapping annotations. We discovered that highlighting of overlapping annotations is challenging but their usability has not been evaluated very much in literature [18]. Thus we developed two approaches for visualizing overlapping annotation and implemented them as a HTML/JavaScript webpage[2]. In the following we describe the user studies and discuss their results.

3.2.1 Related Work

Automatic annotation systems are predominantly implemented as services without a user interface (e.g., [49, 15, 53]). Most manual annotation tools available do not provide references to semantic identities. Visualization of atoms and annotations then becomes the predominant characteristic; significant properties are cardinality (between atoms and annotations), atom granularity (e.g., word, phrase or sentence), and positioning (e.g., handling of overlapping and adjacent atoms). We briefly summarize our findings and refer the reader to the detailed

[2] In collaboration with Annika Hinze, University of Waikato, New Zealand.

analysis reported in [39].

Almost all tools highlight atoms by assigning a background color [12, 23, 6]. Only few change the font style (e.g., [23, 6, 50]) or add graphical elements (e.g., icons) [23, 14]. In contrast, no common approach for indicating overlapping atoms could be identified; the most frequent techniques are mixed background colors (e.g. [23, 6, 12] and mixed font styles (e.g.,[23, 6]). However, only few tools provide a clear visualization of overlapping atoms. The main problem is distinguishing overlapping atoms of the same category of annotation (as they typically use the same style).

The examined tools usually apply similar visualizations to annotation and corresponding atoms. Additionally, a mouse-over effect typically highlights corresponding annotations and atoms. All tools position annotations near the related atoms (where possible). To the best of our knowledge, none of the tools and annotation interfaces have been evaluated for their ease of use (beyond a simple study in [20]).

3.2.2 User Studies on Visualizing Annotations

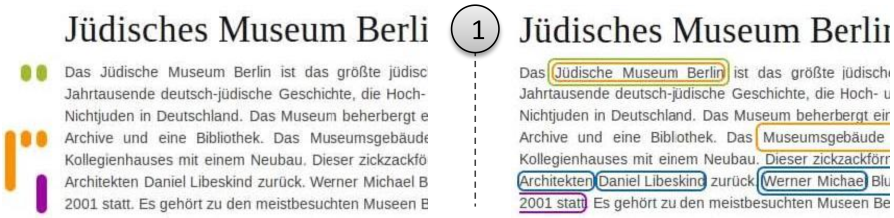
In our user studies we focused on assigning categories to text atoms and the visualization of overlapping annotations. For the purpose of evaluating these two aspects of semantic annotations, we implemented a browser-based system providing two alternative visualizations[3]. The design of the layout is based on our evaluation of existing annotation tools [39]. The main use case of this system is the creation of category-based semantic annotations in texts, where annotations may span a few words or several lines. Our prototype supports a number of pre-defined categories that each have up to ten subcategories.

Study 1: Bar layout vs. border layout

We explored the two alternatives of bar layout and border layout. We realized them as simple prototypes using HTML and JavaScript. In the bar layout, each atom within the text is indicated by a vertical bar in the left margin (Figure 3.4 left). The bars are ordered by length and order in the text. Atoms in the text are highlighted by a mouse-over of the corresponding bar and the annotations appear as a speech bubble near the atom. The border layout highlights annotations by enclosing an atom in a colored frame (Figure 3.4 right). In each layout the color used for highlighting reflects the annotation concept (e.g., orange=architecture, purple=history) and the annotation appears as a speech bubble. Both layouts allow for many-to-many relationships between atoms and annotations, and for atoms to span several lines. The number of atoms overlapping the same portion of text was restricted to three and the number of categories to four.

We observed 12 non-expert participants (P1 to P12) interacting with both interfaces. They started alternatively with bar or border layout. We considered them as non-experts because they were not familiar with semantic technologies (cf. Figure 3.5). All participants are computer literate and $\underline{P}3$, $\underline{P}5$, $\underline{P}7$, $\underline{P}8$, $\underline{P}9$, $\underline{P}10$ and $\underline{P}12$ are non-experts with regards to annotations (tagging+annotations < 5). Throughout the paper, we visually indicate expertise thus: $\underline{P}$x and $\bar{P}$x.

[3] The fully-functional prototype is available online at http://www.corporate-semantic-web.de/overlapping/



Jüdisches Museum Berli

Das Jüdische Museum Berlin ist das größte jüdisc
Jahrtausende deutsch-jüdische Geschichte, die Hoch-
Nichtjuden in Deutschland. Das Museum beherbergt e
Archive und eine Bibliothek. Das Museumsgebäude
Kollegienhauses mit einem Neubau. Dieser zickzackfö
Architekten Daniel Libeskind zurück. Werner Michael B
2001 statt. Es gehört zu den meistbesuchten Museen B

Jüdisches Museum Berlin

Das Jüdische Museum Berlin ist das größte jüdisch
Jahrtausende deutsch-jüdische Geschichte, die Hoch- u
Nichtjuden in Deutschland. Das Museum beherbergt ein
Archive und eine Bibliothek. Das Museumsgebäude
Kollegienhauses mit einem Neubau. Dieser zickzackförr
Architekten Daniel Libeskind zurück. Werner Michae Blu
2001 statt Es gehört zu den meistbesuchten Museen Be

Figure 3.4: Bar layout and border layout

During a learning phase, participants familiarized themselves with the system using a short practice text until they were confident about using the user interface. During the application phase, they executed a number of annotation tasks on a longer text. The participants were encouraged to think out loud as they were making decisions in interaction with the prototype. Each study concluded with a guided interview.

Results & Discussion All 12 participants found it easy to select text atoms for annotation. To support users in creating meaningful atoms the system restricted selection to whole words only. However, some participants (P̲3 and P̲5) wished to select parts of composite words, e.g., 'Libeskind' as part of "Libeskind-Bau." P̄2 and P̲10 liked to establish a link between atoms (e.g., the name 'Daniel Libeskind' and the profession 'architect').

The participants seemed to prefer the border layout (see green diamonds in Figure 3.6). Participants who started with the bar layout showed a clear preference for that design. Participants who started with the border layout seemed to like both layouts.

In cause of its clarity most of the participants appreciated the box layout for visualizing text atoms, even for those spanning several lines. Three participants commented on the number of boxes (P̄2 and P̲3): "[clarity] depends on the number of boxes"). During the interview five participants mentioned that they liked the box layout because they can easily relate text atom and annotation content, e.g., identify the category of a text atom. However, three participants stated that the layout becomes less clear if the text contains many annotations

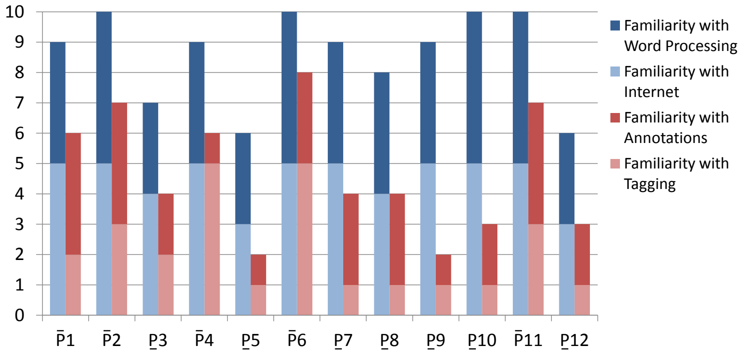


Figure 3.5: Participants' self assessment

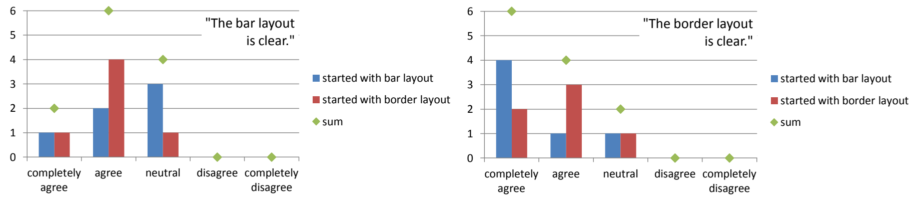


Figure 3.6: Study 1: Clarity of layout

or several categories are assigned to a text atom. All participants preferred the bars to be ordered by length. Seven suggested ordering largest to smallest, four suggested from smallest to largest; one was indecisive. Participants who preferred ordering largest to smallest argued that it would be easier to identify the lines of text (atoms) belonging to the smaller bars. The other group felt the design was clearer when the longer bars were close to the text.

Interviewing the participants we observed that they saw the bar layout to be more suitable for annotating larger text passages because many (small) bars on the left side potentially make the interface less clear. The bar layout was found to be well suited for reading and annotating since texts themselves do not contain any highlighting. Participants found the border layout to be more suited for annotating short text passages because they could easily recognize the atoms, and the relationship between atoms and annotations was clear. However, participants noted that users may get confused by the borders if they are confronted with too many atoms.

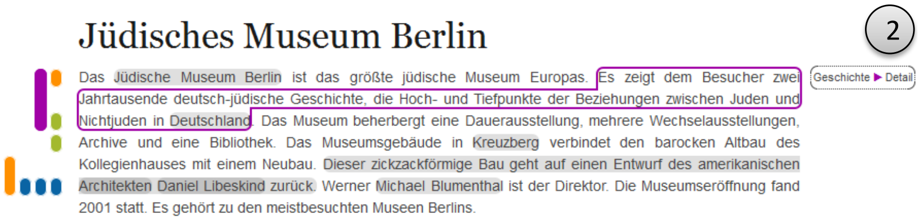
Overlapping annotations constituted a considerate proportion of all created annotations (used by 8 of 12; up to 30 % of all annotations). They were identified as part of a typical annotation process and should not be treated as special cases.

Conclusions for Study 1. We concluded that users seem to be accustomed to the task of assigning categories to text passages. Furthermore, we found that systems should provide a view on the text with two characteristics: (1) a clear view on the text for unhindered reading, a quick overview of the text and locating atoms and annotations at a glance (e.g., bar layout) and (2) detailed information about the annotated text passages for creating annotations (e.g., border layout).

Study 2: Mixed layout

To verify our conclusion we developed a new user interface prototype combining the properties of the bar and border layout (Figure 3.7). We extended the bar layout to highlight the annotated text passages with a light-gray background color. On a mouse-over of a bar the corresponding text is highlighted in the color of the bar using the box from the box layout. Overlapping annotations were indicated by darkening the gray background color.

In this study we observed and interviewed eight participants. The study structure was similar to Study 1: participants had a learning phase for the new layout, an application phase, and a guided interview. All of them also participated in Study 1; we continue using $\bar{\mathrm{P}}$1–$\underline{\mathrm{P}}$12 as references for the participants (omitting $\bar{\mathrm{P}}$4, $\bar{\mathrm{P}}$6, $\underline{\mathrm{P}}$7 and $\underline{\mathrm{P}}$10).



Jüdisches Museum Berlin

Das Jüdische Museum Berlin ist das größte jüdische Museum Europas. Es zeigt dem Besucher zwei Jahrtausende deutsch-jüdische Geschichte, die Hoch- und Tiefpunkte der Beziehungen zwischen Juden und Nichtjuden in Deutschland. Das Museum beherbergt eine Dauerausstellung, mehrere Wechselausstellungen, Archive und eine Bibliothek. Das Museumsgebäude in Kreuzberg verbindet den barocken Altbau des Kollegienhauses mit einem Neubau. Dieser zickzackförmige Bau geht auf einen Entwurf des amerikanischen Architekten Daniel Libeskind zurück. Werner Michael Blumenthal ist der Direktor. Die Museumseröffnung fand 2001 statt. Es gehört zu den meistbesuchten Museen Berlins.

Figure 3.7: Study 2: Combined border and box layouts

Results & Discussion Seven of the eight participants found it 'very easy' and one found it 'easy' to identify the annotated text passages. P̄6 said "[the layout of annotations is] very good, also the Rey background and the information given through mouse-over." All participants stated that it was 'very easy' to identify which category has been assigned to a text passage. About the acceptability of the gray atom background, P̲5 explained "If you look at the text it's only gray. You can't directly see what category is assigned to the text, but that's fine with me."

When asked specifically about ease of identification of long and short atoms, seven participants found them easy (6) or very easy (1) to distinguish (one neutral).

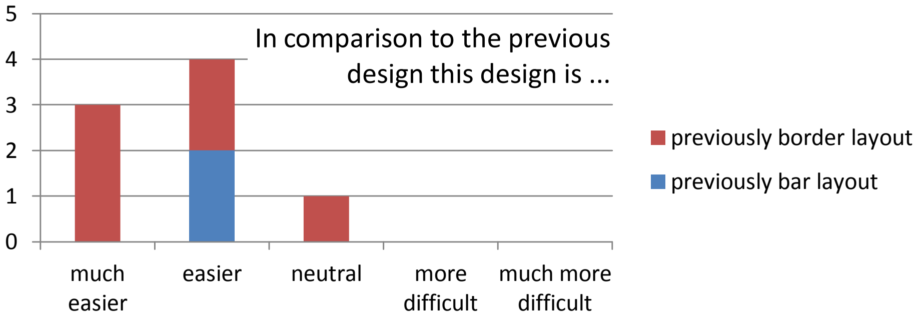


Figure 3.8: Study 2: Clarity of layout

Most participants found the annotation process easy. P̄6 rated it as "neutral" and explained that he meant that the layout was "as good as last time." Five participants (P̄2, P̲3, P̲5, P̄6, P̲12) were surprised about the question (P̲12:"Is there a difference!?", P̲5: "I think the is no difference at all!"). We understand this to mean that the handling of the annotation process itself (not the visualization) felt the same. When asked to compare the mixed layout with their favorite previous one, the participants found it much easier or easier (see Figure 3.8). P̄1 said: "Wow, the new one is much better!"

When asked about what they particularly liked or disliked about the mixed layout in comparison to the previous ones, five participants named the gray highlighting as an improvement, three named it as making the reading harder. Four participants named the mouse-over text to identify the category, while two found the mouse activity "stressful" and the mapping not clear enough "when just looking". P̄6 noted that "the text was more readable in comparison to the border layout.” Several participants particularly mentioned that they liked the combination of bar and border view ($\bar{P}$6: “[it] draws from the advantages of both border and bar view”).

Conclusions for Study 2. Both annotation experts and non-experts were able to successfully create annotations. Even a change of interface and visualization did not influence the participants’ ability to create concept-based annotations. Recognizing existing annotations (as necessary for as post-processing of automatic annotations) has given mixed results – some participants liked the new layout, others felt it still needed improvements (e.g., for easy recognition of annotation categories without mouse-over).

3.2.3 Conclusion

In [18] we discovered that overlapping annotations are a typical use case for annotation tools. In our user study we compared two approaches for visualizing annotations. Especially, we focuses on a clear visualization of overlapping annotations. Depending on the use case and the lengths of the annotations different layouts are more suitable. For example, the bar layout is more suited for reading-focused applications while the border layout is better for identifying the annotation. In our last user study we found that a combination of the bar and border layouts is a good compromise. In our future work we will integrate the layout used in the second study in to the loomp annotation tool. We will also combined it with a new user interface for selecting annotations and semantic identities and redo the loomp user study.

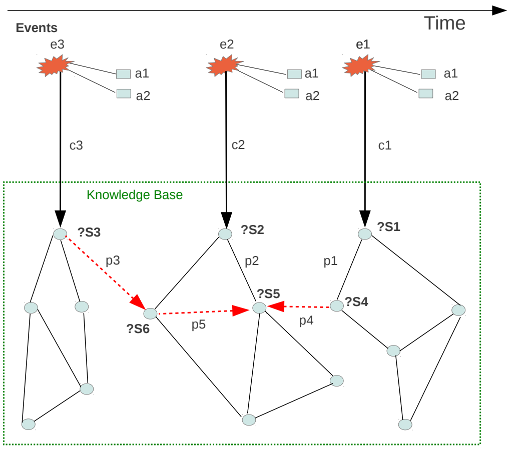
3.3 Semantic Complex Event Processing (AP 5)

Previously, we proposed in [47, 46] a new approach for semantic enabled complex event processing (SCEP). We proposed that the usage of the background knowledge about events and other related concepts can improve the quality of event processing. We described how to formalize complex event patterns based on a logical knowledge representation (KR) interval-based event/action algebra, namely the interval-based Event Calculus [24, 25, 29].

The fusion of background knowledge with data from an event stream can help the event processing engine to know more about incoming events and their relationships to other related concepts. We propose to use one or several *Knowledge Bases (KB)* which can provide background knowledge (conceptual and assertional, T-Box and A-Box of an ontology) about the events and other non-event resources. This means that events can be detected based on reasoning on their type hierarchy, temporal/spatial relationships, or their relationship to other objects in the application domain.

3.3.1 Event Query Rules and Their Categories

Event query rules are declarative rules which are used to detect complex events from streams of raw events. The aggregated knowledge from event streams and background KB can be queried by different types of event queries. These event queries have a hybrid semantic, because they use event operation algebra to



```
{ { (?e1, c1, ?s1) .
    (?s1, p*, ?s) . }
% SEQ event algebra operation
[?e1 SEQ ?e2][Within 5 min.]

 { (?e2, c2, ?s2) .
   (?s2, p*, ?s) . } }

[?e2 SEQ e3] [Within 5 min.]

 { (?e3, c3, ?s3) .
   (?s3, p*, ?s) . } }
```

Figure 3.9: Relation of Events to Resources in the Background Knowledge and Pseudocode of Event Detection Pattern

detect events and they use SPARQL queries to include background knowledge about these events and their relationships.

Lets consider an Event type E_1 which can be instantiated with n (attribute, value) tuples like: $e_1^1((a_1, v_1), \ldots, (a_n, v_n))$. The figure 3.9 shows the event stream and the relationships of events to resources in the background knowledge. An event instance e_1 can be connected to one or more resources in the background knowledge by using a connecting predicate c_1 using one or more attribute value pairs of the event instance.

Our event query rules allow simple event algebra operations, similar to Snoop [11] (i.e. event operations like AND, SEQ, OR, NOT), to query the event stream as well as higher interval-based event operations like (BEFORE, MEETS, OVERLAP, . . .). Our event query rules can include SPARQL query predicate to query external KBs. The results of SPARQL queries are used in combination with event stream to detect complex events. This means that a complex event pattern is defined based on the event operation algebra in combination with SPARQL queries (basic graph patterns plus inferencing on knowledge graph).

One event detection pattern of the relationship shown in the figure 3.9 can be represented by the given pseudocode. The event e1 is connected to the resource s1 in the background knowledge by the predicate c1. In the same way the event e2 is connected to the resource s2 by predicate c2. The predicate p4 connect the two resources s4 and s5, so that it connects the two sub-graphs.

In this report, we describe the most important and interesting categories of event query rules. This categorization is not a complete classification of all possible rule combinations, our aim is more to emphasize interesting rule combinations which can be processed using different event processing approaches. Our implementation of these event query rules and our initial experiments with these rules are described in [48].

Category A - Single SPARQL Query: In this category, the event query rule includes only one single knowledge query and uses its results in one or more variables within the event detection rule. A SPARQL query is used to import knowledge about event instances or types. One or more attributes of events are used to build the basic triple pattern inside the SPARQL query. Category A event query rules can be categorized into three subcategories:

Category A1 - Raw SPARQL: This category of event query rule is the simplest form of these event query rule. The included SPARQL query is only about the resources in the background knowledge. The background knowledge query is independent from the event stream, however the complex event detection is defined on the results of this query in combination with the event stream. In some cases, on each event the SPARQL query should be resent to the KB to update the latest results from the KB.

Category A2 - Generated SPARQL: In this category of event query rules with each incoming event a different SPARQL query is generated and sent to the target knowledge base. The attribute/values of an event instance are used to generate basic triple patterns of a SPARQL query. Based on user definitions some of the tuples (attribute, value) of an event instance are selected and used to generate a single SPARQL Query.

Category A3 - Generated SPARQL from Multiple Events: The query is similar to A2, but the SPARQL query is generated from multiple events. Within a data window (e.g., a sliding time window) from two or more events a single SPARQL query is generated. Multiple events are used to generate the single SPARQL query, the event processing waits for receiving some new events and then generate a SPARQL query based on the emitted events, and query for the background knowledge about them.

Category B - Several SPARQL Queries: Queries of this category include several SPARQL queries and combine them with event detection rules. This means that several A category rules are combined together which can build a category B. The category B of rules are able to combine results from KBs with events using event operation algebra.

Category B1 - Several SPARQL Queries in AND, OR and SEQ Operations: The category B1 is based on the category B, but the results from the SPARQL query predicates are combined with AND, OR, SEQ or similar event algebra operations. The whole query is evaluated on sliding windows of event streams. The SPARQL query predicates are not depending on each other, i.e., the results from one is not used in another SPARQL predicate, so that they are not depending on the results of the other SPARQL query.

Category B2 - Chaining SPARQL Queries: In category B2 several SPARQL queries are generated and executed in sequence. They can be generated based on the results of the previous SPARQL query. Each SPARQL query can be generated from a set of events (e.g., included in a slide of event stream by means of a sliding window, a counting or timing window). This means that different data windows can be defined to wait until some events happened and then a SPARQL query is executed. SPARQL queries might be defined in a sequence chain. The results are directly used for event processing or used in another following SPARQL query.

Category B3 - Chained and Combined SPARQL Queries: In this category SPARQL queries are used in combination with all possible event algebra operations like, AND $\bigwedge$, OR $\bigvee$, SEQ $\bigoplus$, Negation $\neg$, etc. The event operations are used for combining the results from several SPARQL queries or several SPARQL queries are used in combination with event algebra operations like: $((sparql_1 \bigoplus sparql_2) \bigwedge sparql_3 \bigvee \neg sparql_4)$. This category of event query rules is the general form of queries and has the highest possible complexity, because the results from external KBs are used in combination with event operations or the attribute/values from incoming events are used for generation of complex SPARQL queries.

3.3.2 Experiments

For our experiments we required two kinds of data, 1. live real world data from stock market and 2. background knowledge about these events. We used two data sources: background knowledge about companies from DBpedia and the live event stream from Yahoo finance [4]. In addition to these two data sources, we needed to have some mapping between resources in these two data sets. We manually created a mapping hash list between stock market symbols to the URL resources of these companies, for example by searching the DBpedia URL for stock market symbol:
“MSFT” − > “`http://dbpedia.org/resource/Microsoft`”

We have set up two machines, one for an external knowledge base, and another one for the main event processing. Our two machines have Quad Core Intel(R) Xeon(R) CPU E31245 @ 3.30GHz with 16 GB RAM and Debian Linux kernel x86_64 3.0.0-16. The two machines are connected by a dedicated 1000 Megabit/s (Gigabit-Ethernet) LAN.

We installed on one machine Virtuoso Triple store[5] and deployed a complete mirror of DBpedia dataset (version 3.7). The dataset of DBpedia consists of 288 million pieces of information (RDF triples). We configured the virtuoso for the best performance for 8 GB, NumberOfBuffers=680000, MaxDirtyBuffers=500000. The level of reasoning on the dataset is up to RDFS level and below the OWL-lite. Several triple store systems are available which provide different performance and scalability. For our experiments, we only need to use one of them for the comparison of our processing approaches.

For event processing on one of the machines, we used Prova rule engine. Prova can be run as a java application which we used with 2 GB initial and 14 GB maximum java heap size. The highest event processing throughput that we could measure with a simplest event query rule, listing 3.1, is about 450000 events/s (up to 500000 events/s).

```
:- eval(server()).
server() :-  rcvMult(XID, Protocol, From, MSG, {} ),
             sendMsg(XID, osgi, From, reply, {} ).
```

Listing 3.1: Properties for a Company in DBpedia.

The scenario for our experiments is that we have huge amount of background knowledge (288 Million RDF triples) and a high frequency event stream. To the best of our knowledge it is impossible to use one of the existing event processing engines and load such huge amount of background knowledge to the main memory, so that we can compare the existing CEP engines with our approach. However, we compare the improvements achieved by applying different event processing approaches which are proposed in this paper and compare them to each other on the same experimental environment. We have done the following experiments:

[4] http://finance.yahoo.com

[5] http://virtuoso.openlinksw.com , retrieved May 2012

1. In order to know how the performance of doing simple CEP is (SEQ, AND operations on stock market events by normal syntactic CEP processing without any external KBs), we did a normal syntactic event processing to find out the highest throughput of the system. The performance of normal event processing using Prova reactive messaging without any event algebra operations is up to 400000 events/sec. In this experiments, we just receive the event messages(stock market events) and send them them out of the system. Based on the event types and other parallelization parameters of Prova this throughput might be less than 400000 events/sec.

```
:- eval(server()).
server() :- QueryString ='
 PREFIX DBPO:<http://dbpedia.org/ontology/>
 PREFIX DBP: <http://dbpedia.org/resource/>
 ASK { <$url> DBPO:type DBP:Public_company . }',

rcvMult(XID,Procotol,From,Msg,{url->URL}),
sparql_select(QueryString, SID, [url(URL)], 'ENDPOINT'),
sparql_results(SID),
sendMsg(XID,Protocol,From,testrule,{url->URL}).
```

Listing 3.2: A Category A2 Query with polling

2. It is also important to find out the latency of the used KB system (Triple Store). We get a stream of resource URIs of companies and would like to ask if they are public companies or not. This means that we have to generate a SPARQL query for each incoming event and send it to the KB. We performed this experiment on a freshly initialized triple store to be sure that the cache of the triple store is clear. Each SPARQL query can have between 1ms up to 100ms response time depending on the company and the amount of triples for that company. This is also depending on the number of parallel queries on the triple store, we assumed that we have no other system querying the same triple store and we are starting our queries in a sequence.

```
 server():-  rcvMult(XID, P, Sender, event, {url-> URL}),
 server1(URL).

 server1(URL1):-
  rcvMsg(XID, P, Sender, event, {url-> URL2}),
  testrule(URL1, URL2, Industry),
  sendMsg(XID,P,Sender,testrule,{industry->Industry}).

testrule(URL1, URL2, Industry) :-
  Query = 'PREFIX DBPPROP: <http://dbpedia.org/property/>
   SELECT ?i WHERE { <$url1> DBPPROP:industry ?i .
                     <$url2> DBPPROP:industry ?i .
   FILTER(!isLiteral(?i)) }',
  sparql_select(Query,QueryID,[url1(URL1),url2(URL2)],'ENDPOINT'),
  sparql_results(QueryID, Industry),
  retract(sparql_results(QueryID, Industry)).
```

Listing 3.3: A Category A3 Query

3. We have done several experiments to determine the processing performance of each of the query categories. Table 3.1 displays the measured maximum throughput results. For these experiments we came up with exemplary scenarios that are as simple as possible (by using simplest imaginable query). See Listing 3.3 for the category A3 query. In this example, we simply issued another SPARQL request for every incoming pair of messages. Afterwards, we remove the SPARQL results from Prova's internal KB using the *retract* built-in, in order to avoid memory limitations. The listed throughput of category B

Category of query Rules	Throughput (Events/s)
A1 (caching)	280000
A2	2200
A3	1300
B1, B2, B3	≈ 500-4000

Table 3.1: Experimental Performance Results on Different query Rule Categories.

queries should be seen as approximation of maximal throughput results which we have observed during our experiments.

4. In some of the query rules, we have generated and executed SPARQL queries repeatedly based on the incoming events. It is also important to know the performance of rapid execution of SPARQL queries. For each event it sends a query to the triple store and checks the results. Since this is a category A1 query, which includes only a single SPARQL query that does not change over time, the remote triple store can use its internal caching mechanism to speed up the query processing. The highest throughput of this query was about 700 event/s. Since the ***sparql_select*** built-in processes the SPARQL queries synchronously, the throughput is bound by the number of threads used for reactive messaging in Prova and the network latency for communication.

5. For the comparison of the performance improvement by the caching approach and caching the results of SPARQL query in main memory, we have done several experiments to compare this approach with the approach of polling the knowledge base. The throughput of event processing is improved up to 280000 events/s which is about 500 times the throughput of the polling approach. Related to the caching approach, we also conducted some experiments that tested the correlation between the number of results of the (initial) SPARQL query and the throughput.

6. We examined the performance improvement by EQPP approach by simple preprocessing of query rules. For example by importing the results of two SPARQL queries from the KB and creating two simple queries which can be processed without the need to query the external KB. Similar to the previous queries, we created a category B2 query which includes two SPARQL queries and can be rewritten in two separate queries. We manually created two queries from the single one, and we could observe that each query has a maximum throughput of up to 230000 events/s and can be processed in parallel on the same host.

7. Performance improvement of plan-based approach is also evaluated by using query rules which include AND operations between two SPARQL predicates. We examined the performance improvement for the case that we execute the simple SPARQL query first and rather than doing the big one first. We have created a category B2 event query rule which includes two SPARQL queries, Q1 and Q2. The Q1 is a small SPARQL query which has about 2ms answering response time and the Q2 is a large one which has about 20ms answering time. We have manually generated two processing plans for this query. A simplified form of this query is shown in Listing 3.4 with two processing plans. The first plan is to have the large query first, and only if it succeeds the second query should be executed. The throughput of the system with the first plan is about

55 events/sec. In the second plan, we process first the small query, and only if it succeeds the large query is executed. The throughput of this second plan is up to 600 events/s which is about 10 times more than the first plan. In the case that we execute both of the queries in parallel, the throughput is much lower with only around 47 events/sec.

We can see that for some category of rules which are highly dependent on updated knowledge from the KB, steady executing of SPARQL queries might not be avoidable. However, in other categories of rules, the usage of alternative approaches like caching, optimizing the execution plan or query preprocessing can highly improve the performance of event processing. The problem with obsolete query results from the KB can be improved by periodically requesting for updates which does not badly effect the performance of the whole system. Our experiments show also that the scalability of SCEP systems has five different dimensions: 1. discharge rate of raw events, 2. number of rules in main memory, 3. performance of KB (amount of knowledge, query latency) 4. rate of knowledge updates to KB 5. expressive level of reasoning on background knowledge.

```
testPlan1() :-   sparqlrule(QID1),
 server1(QID1), % Find msgs matching Q1.
 sparqlrule(QID2),
 server2(QID2), % Find msg matching Q2.
 sendMsg(XID,Protocol,Sender,testrule,{}).

testPlan2() :-   sparqlrule(QID2),
 server2(QID2), % Find msgs matching Q2.
 sparqlrule(QID1),
 server1(QID1), % Find msg matching Q1.
 sendMsg(XID,Protocol,Sender,testrule,{}).
```

Listing 3.4: Two Different Processing Plans.

3.3.3 Conclusion and Outlook

We described the different categories of event query rules which use special rule predicates for importing data from external KBs and its combination with event algebra operations. For each of rule categories different processing approaches are proposed and are implemented by using Prova rule engine. Our experiments show the potential of the proposed event processing approaches, which can improve the performance and scalability of our knowledge-based event processing system.

Chapter 4

Corporate Semantic Search

Following sections give an overview over the implementation and evaluation of concepts developed in the previous phases of the research pillar Corporate Semantic Search. In our research, we first explored the many possibilities of applying semantic search approaches to the corporate context theoretically and then concentrated on few of them in order to accomplish the proof-of-concept for our approaches. Most of the methods demonstrated below are the result of our work under the industrial cooperation.

4.1 Searching Non-Textual Data (AP3)

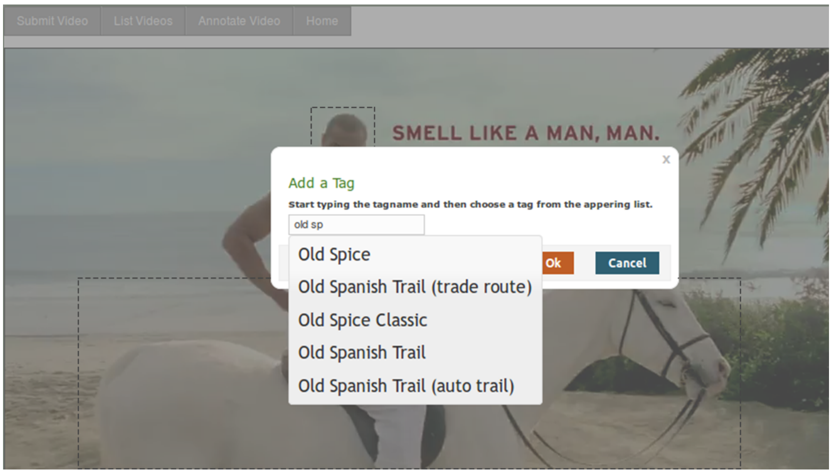
In the last years we've observed an exponential growth of multimedia content on the World Wide Web. However, non-textual content is one of the last frontiers for search technologies and at the same time the most challenging one. Searching inside images or videos has long been deemed impractical due to the processing power required and the need for new and more efficient algorithms. Nevertheless, not even the latest breakthroughs in artificial intelligence and the computational resources large companies have at their disposal have yet yielded what we can call an efficient video search at the web scale. One of the main challenges multimedia search engines are posed with, is the need to efficiently recognize objects in images and video sequences. Machine learning algorithms can achieve this to a certain degree, but fail when a high level of detail is required. For example, it is possible to recognize people walking on a street, but it is a lot harder to recognize what jacket or what brand of shoes they wear. Use cases such as these are especially interesting for companies that look to monetize the advertising possibilities offered by contextual advertisement placement. In [34] we introduced an system 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[19] and machine learning approaches, in order to overcome some the problems faced by existing multimedia search systems. In the first part of this section, entitled "Video Annotation", we describe and evaluate our prototypical implementation in regards to the specific use cases of our corporate partners. In the second part of this section, called "Presentation Slides Annotation", we present a proof of concept semantic presentation slide generator, that we developed in order to test different concepts for manual semantic annotation of multimedia data.

4.1.1 Video Annotation

Video files are traditionally the most complex form of multimedia to process and extract useful information from. In our research project we tried to accomplish this by levareging crowdsourcing approaches coupled with semantic annotations and machine learning.

The Annotation System

One of the approaches we used in order to extract information from non-textual files is that of manual annotations. Our system allows users to pick an object of interest and to annotate it by selecting a tag out of a predefined ontology. That ontology is either an application, or domain ontology that has been purposely built for a specific annotation task, or a world-view ontology such as DBpedia or Freebase. During our research we have observed two very different annotation scenarios. The first consists of collaborative social annotations, where users interact directly with the system and annotate objects based on their interests. Another scenario is represented by automated crowdsourcing systems, such as Amazon Mechanical Turk, where "workers" are paid for annotation tasks. Both scenarios present unique challenges, however after exploring both approaches we focus our work on the second scenario since it is the only one that can be easily be employed by companies, and thereby monetized.



Add a Tag

Start typing the tagname and then choose a tag from the appering list.

old sp

Old Spice
Old Spanish Trail (trade route)
Old Spice Classic
Old Spanish Trail
Old Spanish Trail (auto trail)
Ok
Cancel

Figure 4.1: Multimedia Annotation Interface Prototype

In figure 4.1 we see a prototype of the annotation interface where a user can annotate a shot from an advertising clip by drawing a bounding box around an object and then selecting a concept from an ontology or knowledge base. In this case we use DBpedia as our knowledge base and provide an autocomplete feature in order to improve the usability of the annotation step.

Evaluation

After running our first experiments we came to some noteworthy conclusions:

- The general aspect of the interface is unimportant since it is decided by the general guidelines of the API provider and we have little to no influence on it. This is a big difference to social annotations where users are asked to annotate images without financial motivation.
- Since the motivations of the users annotating the objects are different, we had to introduce quality assurance criteria that ensure reliable results in the annotations.
- Considering the fact that most of the users that create annotations on such crowdsourcing systems are from non-English speaking countries or have poor English skills, we needed to adapt our system in order to be language agnostic.

We tried to make our system more language independent by making use of the advantages offered by Linked Data resources like DBpedia. In these semantic knowledge bases we can find labels in multiple languages for most of the concepts in our ontology. For concepts that do not have a label in the desired language we used Google Translate [1].

We also observed that for crowdsourced annotations based on API systems such as Amazon Mechanical Turk [2] or the evaluation requirements are very different from those of social annotations.

In order to improve the quality of the annotations produced by our system we tried various approaches. [42] presents 3 different aspects to quality assurance: a) Clearly stating the annotation task and making sure the users understand it b) Detection and prevention of cheating attempts and c) Cleaning up errors.

The paper also describes 3 strategies to deal with these problems:

- worker consensus: namely collecting annotations for the same objects from multiple users, and only validating an annotation when multiple users agree on one annotation
- worker control: creating a gold standard with pre-annotated images where the annotations are known to be of good quality, the workers are then presented with some of these images in order to annotate them. If a worker miss-annotated them frequently we can deduce that he is either incompetent or cheating
- worker grading: creating a separate grading task where workers grade the annotations of other workers

Experimental results

We ran an experiment with where we took a sample of 10 preselected video clips. Since annotating each frame would inquire a high cost, we used shot detection and face recognition to significantly reduce the number of frames needed to be annotated. We then submitted a HITs (human intelligence task) to Amazon

[1] http://translate.google.com/

[2] Amazon Mechanical Turk: https://www.mturk.com/mturk/welcome

Mechanical Turk where "workers" were required to annotate the clothing people wear. In the annotation process they were limited to a predefined clothing ontology. Furthermore the workers were required to also specify the material the people were wearing. In order to create a gold standard to evaluate our results we annotated the images ourselves with the best possible attention to detail.

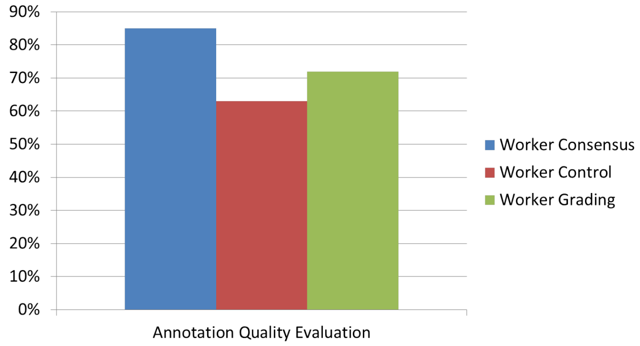


Figure 4.2: Comparison of Quality Control Approaches

In figure 4.2 we compared the results of our annotation experiment with one of the quality control measures to our gold standard. The worker consensus approach yielded the best results even though it increases the cost of the annotations. In this approach 85% of the clothing objects in the videos were annotated correctly. This higher accuracy was due to the fact that we only accept an annotation as valid if at least 2 of 3 workers produce the same annotation. If we want or need to to further increase the accuracy we can choose a 4 out of 5 approach, or combine this approach with the other 2 approaches. The other approaches resulted in lower accuracies and only yielded 63%, respectively 72% accuracy, compared to our gold standard.

The Search System

In order to demonstrate the added benefit of adding semantic annotations to images and video we build a proptoypical search system. However, we note that our scope was not to build a better video or image search system but to build a crowdsourced annotation tool that would enable companies to better monetize multimedia content by optimizing the placement of advertising and complementary information.

The system consists of a simple search interface where a user introduce a search query and is presented with a list of results consisting of multimedia files related to the search query. In order to offer a semantic search system for multimedia content we make use of the research results developed in a previous work package [3] . In contrast to keywords based systems, our system works by matching the search query to the semantic annotations and performing a semantic search as described in the Museumsportal Use Case.

[3] http://www.corporate-semantic-web.de/applications-and-use-cases/articles/museumsportal-berlin.html

Evaluation

Since multimedia content is inherently different from textual content and users react different to it we needed to perform a new evaluation, separate but building upon the evaluation of the previous semantic search work packages.Evaluating semantic search approaches, and search engines in general is a complex task due to the fact that it is prone to subjectivity.

The approach we took in order to evaluate our system to traditional keyword based systems is to do a side-by-side comparison similar to the "Bing It On" challenge that Microsoft used to compare the Bing.com search engine to Google[4]. For our evaluation we generated a test dataset composed of 10 videos of fashion shows. We the proceeded to split the dataset in 2 different sets containing all the 10 videos but differing in the amount of annotation information for each video. One dataset consisted of videos annotated only with the video metadata and textual descriptions of the videos, and the other one included that information and added scene-level semantic annotations. We then proceeded to implemented 2 search systems, a traditional keyword based search engine that uses the first data set and the semantic search engine we described previously. For our test group we selected 15 people and asked them to perform various queries for clothing items such as "red jacket" or "blue jeans". The searches results were presented simultaneously on both systems, with identical interfaces and a side by side display.

At the end of the users interaction with the search systems, after they have tried out various queries which they found interesting, the users were asked which search engine they preferred and why. Out of 15 users 12 preferred the semantic approach. In discussions with the users we found out that they liked the fact that they could find results at a scene or shot level in addition to various general benefits of semantic search such as resolving synonyms, homonyms, aliases, misspellings and performing query expansion. What we found out when interrogating the users that preferred the keyword based approach was that they disliked the fact that one system presented more results than the other. This is due to the fact that the semantic annotation approach presents more results due to the different scenes contained in a single video and the users just wanted to watch the entire video.

4.1.2 Presentation Slides Annotation

One of the byproducts of a corporate environment consists in a large amount of data that is stored in complex and/or proprietary file formats. This information can be valuable for companies but is hard to index and process for search engines and internal document management systems. Furthermore, these presentations often contain other multimedia filetypes such as photographs, charts, sound or video files. These embedded multimedia files are almost impossible to find once placed into a presentation since in most cases there is no associated metadata.

In order to address these problems and create a presentation slide system that would allow companies to efficiently index and reuse the information stored in presentation slides.[22]

[4] http://www.bing.com/blogs/site_blogs/b/search/archive/2012/09/06/challenge-announce.aspx

One big problem of presentation file formats is their complexity, since they try to contain the information as well as the layout and formatting information they become too complex. This complexity makes it impossible for 3rd party programs to efficiently process and index the information contained in the aforementioned file formats. In order to overcome the necessity for dealing with layout and formatting information, we chose to use existing standards that are developed for this specific purpose, namely HTML5. This choice allows us to create presentation slides that can be rendered similarly in all major browsers. Furthermore, choosing HTML5 instead of a proprietary file format makes it possible for 3rd party applications such as web crawlers to parse this information efficiently.

Another problem we want to address is the lack of explicit semantic annotations in the slides. One of the main reasons why slides are so hard to index is the lack of any kind of clear structure or semantic metadata that would tell us what a specific slide or parts of a slide is about. To overcome this problem we allowed the users of our slide generator to explicitly annotate concepts in their presentations with tags from an Ontology. Furthermore, we also introduced automatic annotation functionality in order to increase the usability of the system.

Proof of Concept Implementation

The basic architecture of our implementation consists of a frontend that allows users to create presentation slides in a web-based editor based on HTML5. The frontend communicates with a backend that uses the RDF framework Jena in order to convert the generated HTML5 to RDF and store it in the TDB RDF store. In order to perform automatic annotation we make use of the Alchemy API webservice, which performs Named Entity Recognition.

Web UI

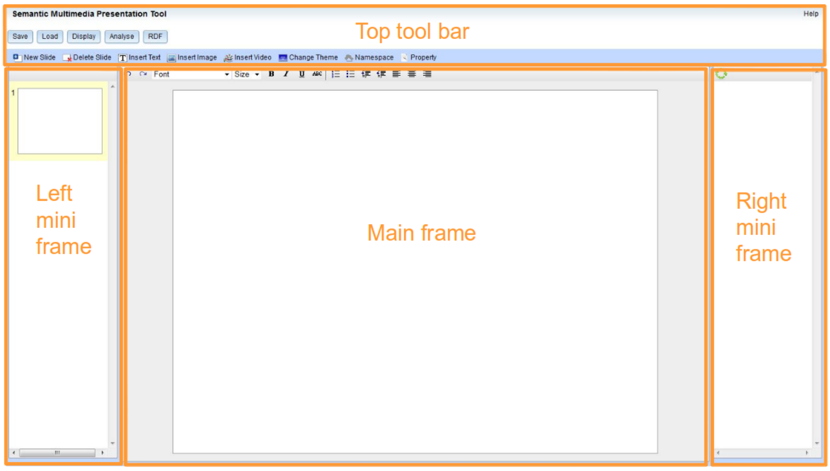
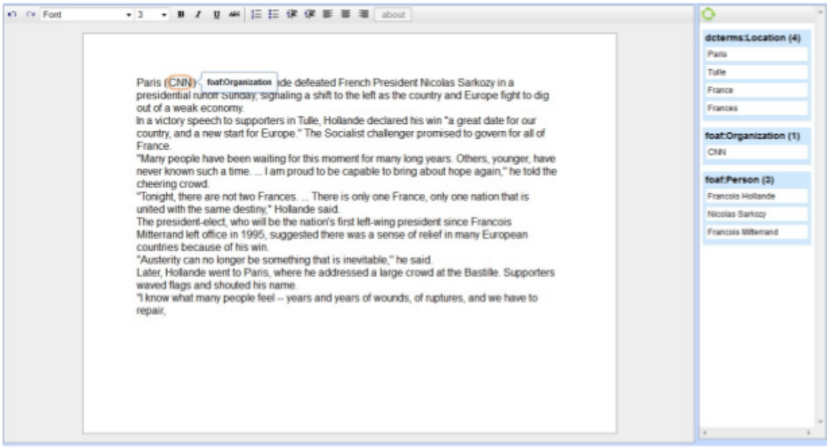


Figure 4.3: Web User Interface

The user interface, which can be seen in figure 4.3, consists of 4 main parts. The top tool bar, which provides basic editing functionality like adding and deleting slides, saving the presentation and so on. Similarly to established slide editors, our tool offers an overview of the existing slides in the left frame,while the main frame shows the current slide and offers editing functionality. Uniquely to our tool however, is the right mini frame that offers information about the semantic concepts in the current slide. These concepts are either manually introduced by the user while he creates the slide or automatically extracted with named entity recognition services.

Text Annotation

Text elements are an important part of presentations, and we have studied text annotation in detail in our Loomp research project[18]. We tried to apply some of the concepts developed in that project here and proceeded to implement manual and automatic text annotation.



Paris (CNN) foaf:Organization ide defeated French President Nicolas Sarkozy in a presidential runoff Sunday, signaling a shift to the left as the country and Europe fight to dig out of a weak economy.
In a victory speech to supporters in Tulle, Hollande declared his win "a great date for our country, and a new start for Europe." The Socialist challenger promised to govern for all of France.
"Many people have been waiting for this moment for many long years. Others, younger, have never known such a time. ... I am proud to be capable to bring about hope again," he told the cheering crowd.
"Tonight, there are not two Frances. ... There is only one France, only one nation that is united with the same destiny," Hollande said.
The president-elect, who will be the nation's first left-wing president since Francois Mitterrand left office in 1995, suggested there was a sense of relief in many European countries because of his win.
"Austerity can no longer be something that is inevitable," he said.
Later, Hollande went to Paris, where he addressed a large crowd at the Bastille. Supporters waved flags and shouted his name.
"I know what many people feel -- years and years of wounds, of ruptures, and we have to repair,

dcterms:Location (4)
Paris
Tulle
France
Frances

foaf:Organization (1)
CNN

foaf:Person (3)
Francois Hollande
Nicolas Sarkozy
Francois Mitterrand

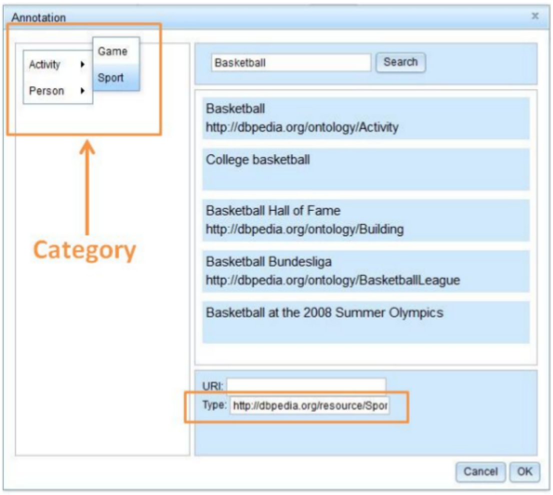
Figure 4.4: Automatic Text Annotation

Figure 4.4 shows the results of the automatic annotation step. The user can activate automatic annotation by clicking on the “Analyse” button in the top tool bar. After this step a series of detected named entities will be shown in the left mini frame. This entities are linked to concepts in ontologies such as DBpedia and Yago and are grouped by categories.

Manual annotation can be done by selecting and then right clicking on a word. A new menu will appear which offers the possibility to add an annotation. Once the user has clicked on the “Add annotation” button a new window will appear where the user can select predefined annotations from an ontology or can search for the corresponding categories in DBpedia. This process is shown in Figure 4.5

Picture and Video Annotation

Since image and video files can describe a wide variety of objects we took a slightly different approach in the way we designed the annotation interface for those filetypes. When a user right clicks a media file he inserted in the presentation he can add an annotation. In the new window that pops up he can



Basketball

Type: http://dbpedia.org/resource/Spor

Basketball
http://dbpedia.org/ontology/Activity

College basketball

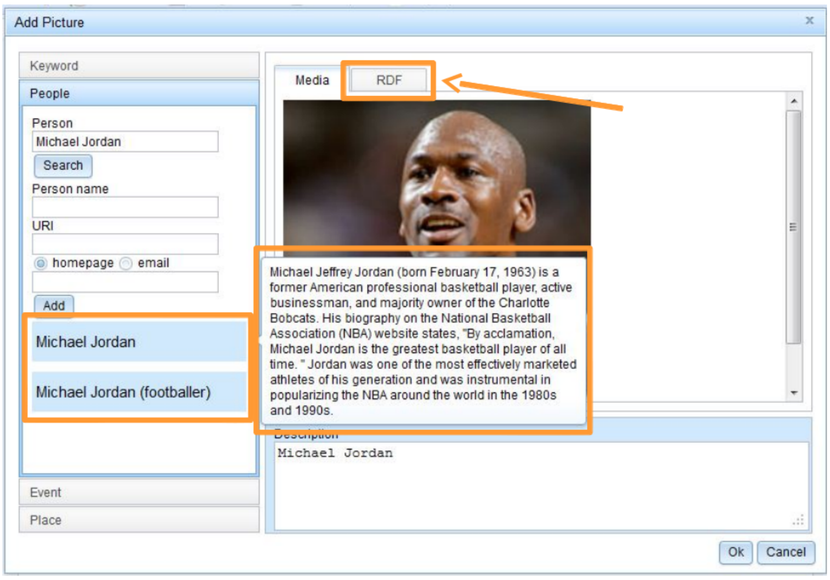
Basketball Hall of Fame
http://dbpedia.org/ontology/Building

Basketball Bundesliga
http://dbpedia.org/ontology/BasketballLeague

Basketball at the 2008 Summer Olympics

Figure 4.5: Manual Annotation

not only select concepts with which he can annotate that media file but also search DBpedia for matching concepts. Furthermore we extract existing metadata from the media files and convert it to RDF. After a user has annotated the media file he can inspect the resulting RDF by clicking on the RDF tab.



Person: Michael Jordan

Michael Jordan

Michael Jordan (footballer)

Michael Jeffrey Jordan (born February 17, 1963) is a former American professional basketball player, active businessman, and majority owner of the Charlotte Bobcats. His biography on the National Basketball Association (NBA) website states, "By acclamation, Michael Jordan is the greatest basketball player of all time." Jordan was one of the most effectively marketed athletes of his generation and was instrumental in popularizing the NBA around the world in the 1980s and 1990s.

Description: Michael Jordan

Figure 4.6: Image and Video Annotation

4.1.3 Conclusion

In the first part of this section we presented a novel method of analyzing video data by using Crowdsourcing services and semantic technologies we demonstrated that companies can make use of semantic annotations in order to better annotate their content. We presented some of the unique challenges posed by crowdsourcing approaches as well as some of the solutions to them. In collaboration with our corporate partners we developed a system that makes use of semantic technologies, machine learning and crowdsourcing in order to annotate and then monetize multimedia content. We then proceeded to demonstrate the value of semantically enriched multimedia content compared to un-enriched content.

In the second part we presented a proof of concept implementation of a semantic presentation slide generator. We focused strongly and the user interface and tried to improve the usability our annotation interface rather than on video processing and crowdsourcing like in the video annotation part. The main conclusion we reached after developing this prototype is that presenting an user friendly interface where people can also annotate their images and text while creating presentations can lead to the creation machine readable presentation slides that can be reused and searched through more efficiently.

4.2 Evaluation of Recommender Systems (AP 4)

With the growing amount of data available in the web, the users are unable to cope with this massive overload of information. They need a support to find the information which fulfills their personal needs. Especially in online-stores, where they are confronted with a lot of products with nearly similar features, they are unable to find the suitable products. Nowadays, when competing stores are only one-click-away this means a real income loss and circumstances which have to be avoided. By using techniques which adapt the behavior of the page to the needs and goals of a particular user, information-portals can keep their visitors longer. This personal adjustment of the page to each single user can be achieved with the so called *Recommender Systems* [13].

Recommender Systems create for each user an user-profile which maps his interests to an internal representation. Based on this profile, a *Recommender Systems* computes the items which best match the personal taste of the user. This can be either done by comparing different user-profiles to each other *(Collaborative Filtering)* [38] or by comparing the extracted features of an item to the stored interest in the user-profile *(Item-Based Filtering)* [36]. Both approaches have specific benefits and drawbacks which make them explicitly suitable to different domains. These approaches can be combined to a *Hybrid-Recommender System* [10]. In the best case, a system can be achieved, which combines the benefits of the particular different single systems but without their drawbacks. Furthermore, additional information about the relations and meanings of the different features can be used in the process of finding the best matching item, which creates a *Knowledge-Based-Recommender System* [9]. When used in combination, for example with *Collaborative-Filtering*, this *Hybrid-Recommender System* can provide far better computed recommendations as the traditional approaches. Nevertheless, this knowledge is deeply integrated into the system and can hardly be maintained, extended or even be applied to a different domain than the original one.

With the growing amount of available information in the *Semantic Web*, more and more knowledge exists in a machine-readable format. Ontologies can be used to provide the needed knowledge-base for a *Knowledge-Based-Recommender System*. While keeping the benefits of this kind of *Recommender System*, ontologies provide a modular and changeable knowledge-base for the *Recommender System*. Furthermore, public ontologies are usually maintained by domain-experts, so their contained information grows and their quality improves over time. A *Recommender System* which uses ontologies as its knowledge-base is called a *Semantic Recommender* [37].

4.2.1 Evaluation

In order to investigate the benefits and drawbacks of the different *Recommender System* approaches, we implemented, in coopertaion with T-Systems Multimedia Solutions, a modular evaluation framework in *JAVA*. When using the same evaluation-data for the different algorithms, a common used framework allows their comparison under an objective point of view. For this framework, several *Recommender* algorithms [5] were implemented and evaluated.

Algorithms

The lowest grade of semantics can be found in the *Collaborative Filtering Recommender*. They only use the ratings of the users and do not regard specific interests of users or the features of items. We chose the traditional *User-To-User Collaborative Filtering* as well as the *Item-To-Item Collaborative Filtering* because of their wide spread use. Recommendations are computed by finding the users which have the same interests and taking their items with the highest ratings in the neighborhood, which the actual user has not rated yet. In *Item-To-Item Collaborative Filtering*, the best items are found by comparing the ratings on them which follows the premise that similar items have usually the same ratings.

The next higher grade of semantics can be achieved with the *Item-Based Filtering*. Each item is represented as a vector of its describing features and each user profile contains a vector the features the user is interested in. Recommendations are found by comparing the features of the items with the feature vector of the user using the common used metrics of Information Retrieval like the Cosinus-Measure [8]. The most similar items are then recommended.

By using external knowledge in form of a taxonomy of features, the simplest kind of a *Semantic Recommender*, a *Taxonomic Recommender* is realized. It not only uses the features of the items and the interest of the user, it regards their hierarchy and relations too. By expanding the item-vectors and user-profiles with hyperonyms (concepts, which are more generic terms of the features and are higher in the hierarchy), items of interest can be found even if they do not share a feature with the already rated items. This can provide more diverse recommendations.

The highest grade of semantics is used in a *Full Semantic Recommender*. It not only uses a taxonomy of features, in addition, the extracted features of the items are stored in an ontology. This ontology forms a network of connected

[5] traditional ones as well as more sophisticated algorithms with external knowledge-bases, with an growing grade of used semantics

concepts and therefore represents their relations between the items and their features. The implemented *Recommender System* uses an approach which is called *Property Propagation* [21]. The similarity between the different features is propagated through the network so that non-obvious connections between concepts and therefore items can be found.

Evaluation Data

As evaluation data, the MovieLens data-set with 1 Million ratings was chosen. This data set contains 1 Million ratings of 4.000 users on 6.000 movies. Not only this data set is based on real data, its public availability justifies its primary choice for evaluating *Recommender System* algorithms. Based on the contained movies in the data set, several other sources for additional data were used. The imdb-website[6] provides tags which describe the content of the movies. These tags are used as features for the item-vectors of the *Item-Based Filtering Recommender System*. The tags were analyzed on their semantic meaning and were then mapped to unique concepts using the *WordNet* lexical database to realize an *Item-Based Filtering Recommender System* with a higher grade of semantics. The hierarchy of these concepts is used by the *Taxonomic Recommender*. Apart from the tag-based taxonomy, another taxonomy derived from the dbpedia[7] has been used as additional input by the *Taxonomic Recommender*. In the dbpedia, each movie has several genres assigned to it. These genres are further divided and classified in a hierarchy with the movie-node as its base concept. Furthermore, the dbpedia is used to gather the necessary ontology for the *Full Semantic Recommender*. All information about a movie and the relations between the data forms the used ontology.

Evaluation-Criteria

The algorithms were evaluated using the *MAE*-Metric [17] as objective measure for the quality of the generated recommendations. This metric measures the difference between a computed recommendation and the already given rating in the evaluation set. When using a large number of user profiles a good comparison between different algorithms can be achieved. Therefore, this measure is widely used in the literature on *Recommender Systems*.

Furthermore, the algorithms have been evaluated according to their learning behavior. Recommendations can only be computed efficiently, when enough information about a user is collected, so that this learn rate is crucial for providers of a *Recommender System*. Especially for *Recommender Systems* which compute the recommendations by comparing user profiles, the number of already stored profiles is important. Only if enough other user profiles are stored, recommendations can be calculated. This fact is evaluated by using only a part of the training data provided by the evaluation data set.

Evaluation Results

For the evaluation, the data set was randomly divided into two smaller sets, a training set which contains 80% of the user profiles and a test set, which

[6] `http://www.imdb.com/`

[7] `dbpedia.org`

contains the remaining 20%. These sets were identical for all evaluation runs. In the first step, the algorithms have been trained with the user profiles from the training set. For each of the remaining user profiles in the test set, the *MAE* was computed by creating a virtual profile consisting of all ratings but one. This was repeated for all ratings of a user and the mean over all profiles creates the computed evaluation result.

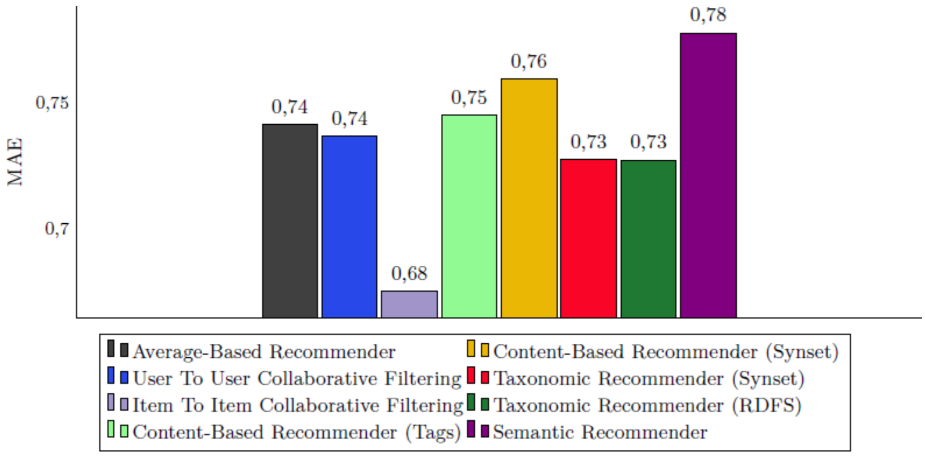


Figure 4.7: Quality of the *Recommender Systems* in respect of the *MAE*-metric

Figure 4.2.1 shows the quality of the different algorithms in respect of the *MAE*-metric. The traditional *Item-To-Item Collaborative Filtering* outperforms the other algorithms, especially the approaches which use some kind of additional knowledge. This fact can be explained by the problems arising by the use case. In a domain, where the extraction of suitable describing data for the items is rather complicated, traditional *Collaborative Filtering* algorithms which only rely on the subjective ratings have a greater performance than the content-regarding algorithms. But the quality of the recommendations computed by the *Taxonomic Recommender* is second in the set of algorithms. This can be justified by the fact, that not all items had a description and the quality could be even better if enough data had been available. For the *Full Semantic Recommender*, this fact applies even in a stronger significance, since this algorithm completely relies on a formal description of the items.

Figure 4.8 shows the learn rate of the *Recommender Systems* regarding the number of ratings. Having the greatest quality of recommendations among the evaluated algorithms, *Item-To-Item Collaborative Filtering* reaches this high quality only after collecting a large number of ratings for a particular user. The *Taxonomic Recommender* needs only a comparatively small amount of information from the user to find the most suitable items for him. This makes this approach highly suitable in systems, where the users do not want to invest much effort in training and updating their profiles.

Figure 4.9 shows the learn rate of the *Recommender Systems* regarding the number of comparison profiles in the system. The implemented *Collaborative Filtering* as well as the *Taxonomic Recommender* generate their recommendations by finding the most similar users to the user for who the recommendation

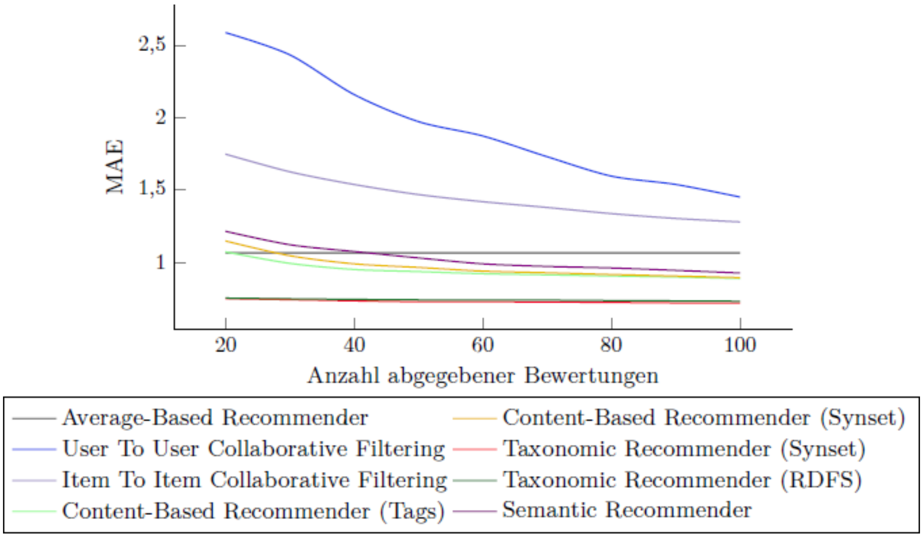


Figure 4.8: Learn rate of the *Recommender Systems* regarding the number of ratings

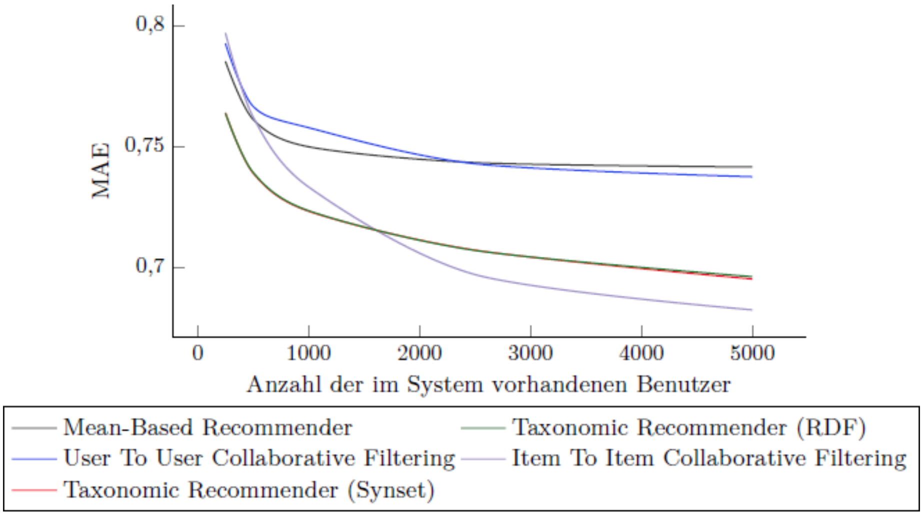


Figure 4.9: Learn rate of the *Recommender Systems* regarding the size of the training data

should be calculated for. Therefore, they need a certain amount of already stored user profiles to find matching other profiles. The *Taxonomic Recommender* outperform the other algorithms because they need a smaller number of stored profiles then the traditional *Collaborative Filtering.* Because of the profile expansion, they can find relations between the profiles even if the different users have no common rated items. Nevertheless, the *Item-To-Item Collaborative Filtering* generates recommendation of a higher quality, once enough data is stored in the system.

4.2.2 Conclusion

Especially the *Taxonomic Recommender* shows an overall better performance than the traditional approaches. Even if the *Item-To-Item Collaborative Filtering* generates recommendations of a higher overall quality according to the used evaluation-metric, the *Taxonomic Recommender* outperforms the traditional *Recommender Systems* in other areas, especially in its learning behavior. It needs less data, to provide recommendations with a good quality. The evaluated *Taxonomic Recommender* could not show its full potential since it worked only on a partial data set. The same problem had the *Full Semantic Recommender*, but even in a higher grade, since his behavior fully depends on a formal representation of the domain. If data for all items had been available, even better results can be expected.

Many items in online-stores are classified in taxonomies and today, these stores use more and more semantic technologies to describe their items. With the growing amount of publicly available data in a formal representation format, the *Semantic Recommender Systems* could be the next step in the evolution of *Recommender Systems.* They overcome several restrictions of traditional approaches and with the additional knowledge they use, they can provide better recommendations.

4.3 Integrated Personalized Views (AP 14)

On top of the syntactic (XML, relational, unstructured) and semantic (RDF,RDFS, OWL) data/knowledge layer, rules play an important role to automatically and contextually transform data and integrate existing knowledge in order to derive new conclusions and decisions. [26] Rules provide a powerful and declarative way to represent and control the manifold personalized views on corporate knowledge. Semantic agents can exploit rules to represent their decisions on how to use knowledge in a particular personalized view. These views define the agent's purposes or goals such as active selection and negotiation about relevant (semantic) meanings, achievement of tasks, and internal and external reactions on occurred events, changing conditions or new contexts. This extends the Corporate Semantic Web to a rule-based *Pragmatic Web* [8] [52] which puts the independent corporate ontologies and domain-specific data of a personalized view into a pragmatic context, such as collaborative situations, organizational norms, purposes or individual goals and values. In this section we will further describe this agent-based approach for personalized views which integrate corpo-

[8] http://www.pragmaticweb.info/

rate data and knowledge by rule technologies and which makes use of corporate (context) ontologies managed in distributed Maven repositories.

4.3.1 OntoMaven - Distributed Maven-based Ontology Project Management

OntoMaven is a set of Maven plug-ins supporting the management and engineering of ontologies from distributed (Maven) ontology repositories. OntoMaven extends the Apache Maven tool and adapts it for distributed ontology engineering and ontology-based software engineering.

OntoMaven dynamically downloads ontologies and plug-ins for ontology engineering from distributed OntoMaven repositories and stores them in a local cache. OntoMaven repositories are extensions of Maven repositories with ontology version and maintenance metadata. Via plug-ins for search and transformation it can also use ontology search engine interfaces and APIs (such as OMG ODM, OMG API4KB, web search engines such as SWOOGLE) to publicly accessible ontology repositories and distributed KBs (given appropriate semantic transformations and interoperations are defined in the search and transformation tasks).

It extents an OntoMaven POM file to describe the ontology project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as:

- Versioning
- Dependency Management for (distributed) Ontologies
- Ontology Documentation
- Testing
- IDEs/APIs tasks for download, transformation-integration/import-compilation, installation and deployment as e.g. ontology service or ontology API

OntoMaven is built using Mavenâs plugin-based architecture that allows it to make use of any application controllable through standard input. Therefore it encapsulates several of the CSW technologies, such as SVont, Concept Grouping for Documentation, Semantic Matchmaking, together with other APIs and IDE functionalties into Mojo PlugIns, such as the SVoNT ontology versioning, the ontology modularization algorithms and concept grouping functionalities in the ontology documentation.

As described in Technical Report TR-B-12-04 [34] an ontological model for representing context information consists of several different vertical and horizontal layers of modular ontologies from top level to domain, task, and application ontologies. OntoMaven supports the management, maintenance, integration and deployment of such modular context ontologies from distributed (extended) Maven ontology repositories.

4.3.2 Concept - Rule-Based Agent Architecture

Rule technologies have been proven to be a powerful declarative knowledge representation formalism which supports a separation of concerns principle. We make use of rules to integrate semantic knowledge in a declarative way into personalized views which are modeled within rule-based computer agents [7, 1]. The major benefits of this approach are that rules can be managed separately from the underlying data and application in which they are used, and that they can be written and dynamically adapted by business users and automatically executed in a rule engine based on the underlying logical semantics. We run such rule engines as inference services within computer agents which are accessible via standardized rule interfaces (see figure 4.10).

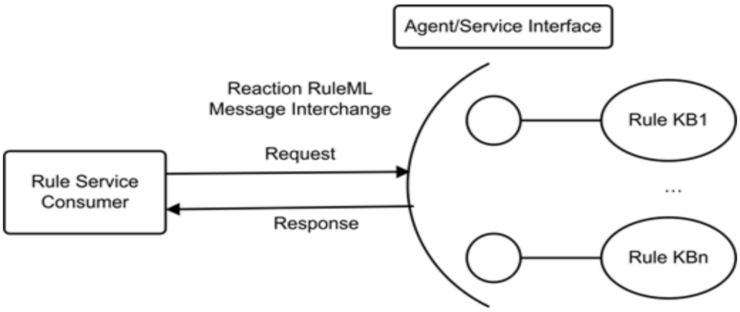


Figure 4.10: Rule Based Agent

The abstraction into computer agents supports the modelling of different roles in an organisation, together with their personalized views, individual contexts, decisions and efforts/tasks. [31] The distributed nature of the agent-based approach allows for modularization, information hiding (e.g. privacy views) and different negotiation and coordination strategies by loosely-coupled (via standardized interfaces) or decoupled (via event messages) interactions between agents.

We have worked on rule language technologies such as Prova and standards such as RuleML, Reaction RuleML, W3C RIF, OASIS LegalRuleML on different layers. [30] On the computational independent layer, rules are are engineered in a controlled natural or graphical language which is easy to use and understand for users. [55] The rules are mapped and serialized in a platform independent rule interchange format such as RuleML [5, 32] or W3C RIF. These platform independent formats support the interchange of rules between different specific rule engines and rule execution environments. From the rule interchange format the rules are translated into a platform specific rule language for execution.

To represent the different aspects of personalized views (see [34]), a certain level of expressiveness is required. The main requirements are:

- different types of rules such as deduction rules, reaction rules, transformation rules, constraints to derive conclusions and decisions from data, to describe reactions and transformations and to proof compliance to constraints.
- built-in data types and query languages to integrate external data and data models into rules

- typed rule logic to support external type systems such as ontologies and object-oriented (Java), which can be used to type the rule terms such as variables

This enables us to reuse the existing corporate ontologies and data models, such as our modular context ontologies described in [34, 45] in the execution and interpretation of the rules. Based on this the rules can derive consistent personalized views from the underlying data and application layers. Figure 4.11 illustrates this interplay of rules and ontologies.

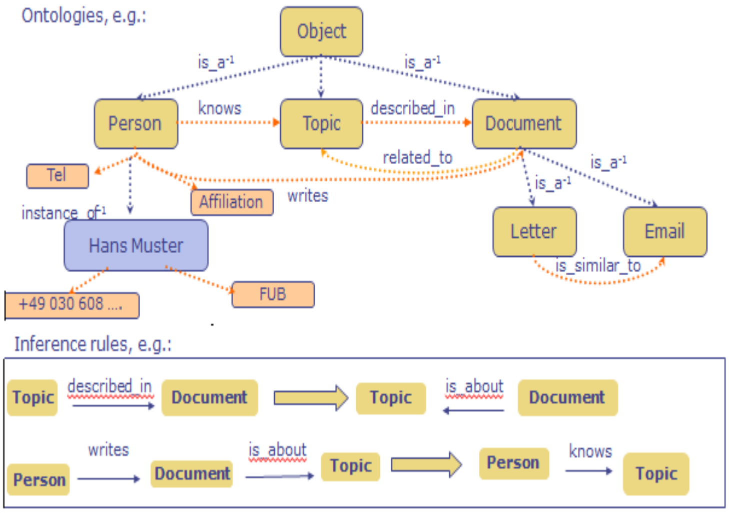


Figure 4.11: Semantic Technologies - Ontologies and Rules

The agent architecture is capable of implementing organizational semiotic structures. A common hierarchical agent topology is, e.g., represented by a centralized star-like structure with Organizational Agents (OAs), which act as central orchestration nodes to control and disseminate the information flow from and to their internal Personal Agents (PAs), and the External Agents/Services (EAs) and internal Computational Agents/Services (CAs). [55]

Organizational Agent

An Organizational Agent (OA) represents a virtual organization (respectively network of agents) as a whole. An OA manages its local Personal Agents (PAs), providing control of their life cycle and ensuring overall goals and policies of the organization and its semiotic structures. OAs can act as a single point of entry to the managed sets of local PAs to which requests by EAs are disseminated. This allows for efficient implementation of various mechanisms of making sure the PAs functionalities are not abused (security mechanisms) and making sure privacy of entities, personal data, and computation resources is respected (privacy & information hiding mechanisms). For instance, an OA can disclose information about the organization to authorized external parties without revealing private information and local data of the PAs, although this data might have been used in the PAs to compute the resulting answers to the external requester.

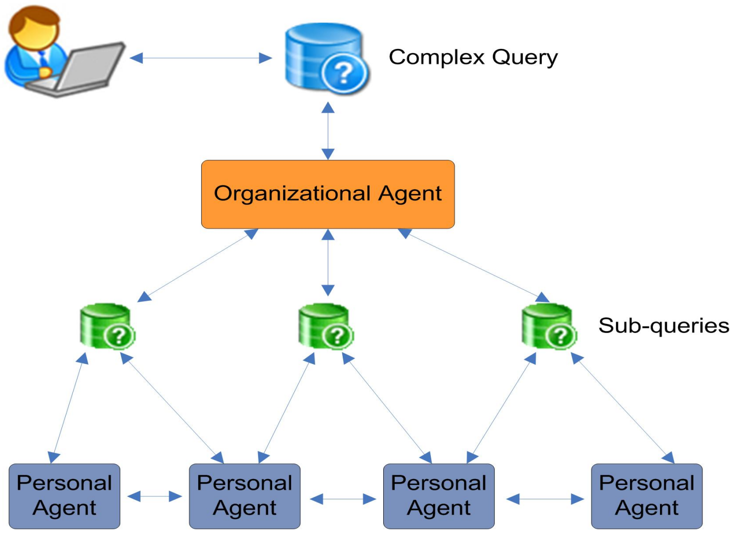


Figure 4.12: SymposiumP

Personal Agents

Personal Agents (PAs) assist the local entities of a virtual organization (respective network). Often these are human roles in the organization. But, it might be also services or applications in, e.g. a service oriented architecture. A PA runs a rule engine which accesses different sources of local data and computes answers according to the local rule-based logic of the PA which represents the personalized views of a PA. Depending on the required expressiveness to represent the PAs rule logic arbitrary rule engines can be used as long as they provide an interface to ask queries and receive answers.

Importantly, the PAs might have local autonomy and might support privacy and security implementations. In particular, local information used in the PA rules becomes only accessible by authorized access of the OA via the public interfaces of the PA which act as an abstraction layer supporting security and information hiding. A typical coordination protocol is that all communication to EAs is via the OA, but the OA might also reveal the direct contact address of a PA to authorized external agents which can then start an ad-hoc conversation directly with the PA. A PA itself might act as a nested suborganization, i.e. containing itself an OA providing access to a suborganization within the main virtual organization. For instance, this can be useful to represent views on nested organizational structures such as departments, project teams, service networks, where e.g., the department chair is a personal agent within the organization and at the same time an organizational chair for the department, managing the personal agents of the department.

Internal Computational Agents

Computational Agents (CAs) act as wrappers around internal computational services and data which is used in the PAs' personalized views. They fulfill computational tasks such as collecting, transforming and aggregating data from internal underlying data sources or computational functions. That is, they integrate external data into the internal knowledge base of the PAs which use their rules to transform the data into knowledge and construct personalized views out of it. The PAs can communicate with the CAs decoupled via interchanging event messages or loosely coupled via the built-ins and service interfaces of the rule engines in the PAs.

External Agents

External Agents (EAs) constitute the points-of-contact that allow an external user or service to query the Organizational Agent (OA) of a virtual organization. The OA answers these queries based on the rule-based views of the personalized PAs. An EA is based, e.g., on a Web (HTTP) interface that allows such an enquiry user to pose queries, employing a menu-based Web form. An external agent can be e.g. an external human agent, a service/tool, or another external organization agent, i.e. leading to cross-organizational communications.

4.3.3 Proof of Concept Implementation

The implemented framework (see figure 4.13) consists of three interconnected architectural layers, listed here from top to bottom:

- Computationally independent user interfaces such as template-based Web forms or controlled English rule interfaces. [55]
- Reaction RuleML [32] as the common platform-independent rule interchange format to interchange rules, events, actions, queries, and data between agents and other agents (e.g., Semantic Web services or humans via Web forms).
- A highly scalable and efficient enterprise service bus (ESB) as agent/service-broker and communication middleware on which platform-specific rule engines are deployed as distributed agent nodes (respective semantic inference Web services). We use Prova [26] as expressive platform specific rules engine to manage and execute the logic-based personalized views of the semantic agents in terms of declarative rules which have access to semantic ontologies.

In the following, the rule-based agent framework will be explained from bottom to top.

Prova [26] is an enterprise strength highly expressive distributed Semantic Web logic programming (LP) rule engine. One of the key advantages of Prova is its elegant separation of logic, data access, and computation as well as its tight integration of Java, Semantic Web technologies, and service-oriented computing and complex event processing technologies. Prova follows the spirit and design of the W3C Semantic Web initiative and combines declarative rules, ontologies and inference with dynamic object-oriented programming.

Prova provides a rich library of built-ins including support for many query languages such as SQL, XQuery, SPARQL, File IO, so that external data sources can be dynamically accessed at runtime and the selected data can be used in the rules to derive more complex personalized views. Prova supports external type

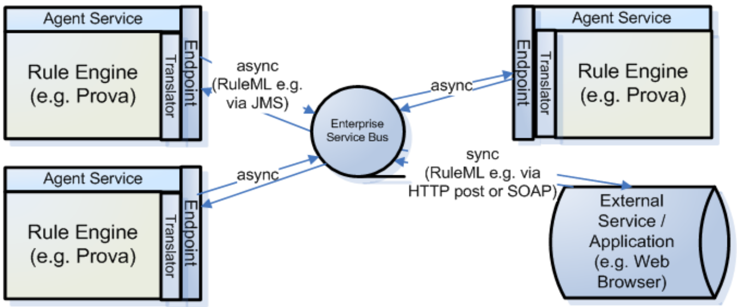


Figure 4.13: Rule Based Agent Framework

systems such as e.g. Java class hierarchies or Semantic Web ontologies (RDFS, OWL) via its typed order-sorted logic. Rule terms, such as variables, can become typed and hence instantiated only by data corresponding to the defined ontological or object-oriented semantics. Furthermore, the ability to embed Java calls directly in Prova rules enables the integration of highly optimized Java based computations and existing (Enterprise) Java Bean functions.

Modularization of the knowledge base is another important expressive feature of Prova which is required to implement different agent roles in the same agent instance (the same knowledge base). It is possible to consult (load) distributed rule bases from local files, a Web address, or from incoming messages transporting a rule base. System-defined as well as user-defined meta data labels can be used to manage the modules in a knowledge base. This label can be e.g. used for asserting or retracting complete modules from the knowledge base and for scoping queries / goals to apply only on the particular module. This is in particular use-full for expressing special policies in personalized views, such as e.g. privacy constraints.

RuleML [5] has been designed for the standardized interchange of the major kinds of rules in an XML format that is uniform across rule languages and platforms. The family's top-level distinction is deliberation rules vs. reaction rules. We use it on the platform-independent layer for the communication between the agents which exchange queries, answer, and rules. Therefore the platform-specific Prova syntax becomes translated into **Reaction RuleML** [32] messages which are transported by the Enterprise Service Bus middleware to the appropriate agent(s).

On the computation-independent level, online user interfaces allow external (human) agents issuing queries or uploading rule sets to the agents (typically the OA) in a controlled natural language, template-driven Web forms and graphical notations. Translation services, such controlled English translators, map into standardized Reaction RuleML messages based on domain-specific language translation rules.

4.3.4 Demonstrator

We have validated our proposed proof-of-concept solution in different application scenarios such as corporate reputation management [28], business processes and workflows [27, 54], event organizations [55], and others [31]. In this subsection we demonstrate the proposed concept and implementation by a concrete demonstrator use case.

Simple views over data can be constructed by built-in data queries. The following rule in Prova syntax uses a SPARQL query to select all manufacturers of luxury car manufacturers from DBPedia (RDF Linked Open Data database extracted from Wikipedia - see DBPedia Deutsch `http://de.dbpedia.org` for more information) .

```
luxuryCar(Manufacturer, Name, Car) :-
    Query="SELECT ?manufacturer ?name ?car   % SPARQL RDF Query
   WHERE {?car <http://purl.org/dc/terms/subject>
                  <http://dbpedia.org/resource/Category:Luxury_vehicles> .
     ?car foaf:name ?name .
     ?car dbo:manufacturer ?man .
     ?man foaf:name ?manufacturer. } ORDER by ?manufacturer ?nameâ,
sparql_select(Query,manufacturer(Manufacturer),name(Name),car(Car)).
```

More complex rule-based views can be implemented on top of such simple data views. The following rule of a computational agent (CA) actively filters for car manufacturer stocks from a real-time ticker feed and enriches the stock ticker event with the queried data from the DBPedia data view. It then sends the enriched knowledge to the personal agent ”epa1”.

```
rcvMult(SID,stream,"S&P500", inform, tick(StockSym,Price,Time)) :-
    % filter stock symbols which are of type car manufacturer
manufacturer(StockSym, Man^^car:Car_Manufacturer),
    % find and create a list of all luxury car manufacturers
findall([luxuryCar|Knowledge],luxuryCar(Man,Name,Car),Knowledge),
    % enrich the stock symbol with the knowledge about luxury car manufacturers
    EnrichedData = [StockSym,Knowledge],
    % send enriched knowledge to personal agent epa1
    sendMsg(SID2,esb,"epa1", inform, happens(tick(EnrichedData,Price),Time)).
```

The following rule of the personalized agent (PA) implements a personalized view. The rule accepts queries from the organizational agent for monitoring luxury car manufacturers. It actively receives the knowledge-enriched stock ticker feeds from the computational agent(s). If the monitored luxury car manufacture is a member of the received knowledge (about all luxury car manufactures listed in Wikipedia) the stock tick is part of the personalized view and will be send to the requesting OA.

```
rcvMult(CID,esb, OA, query, monitor(LuxuryCarManufacturer, Ticker) :-
    rcvMult(SID2,esb, CA, inform, happens(tick([StockSymbol | Knowledge],P),T))
    member(LuxuryCarManufacturer, Knowledge),
    Ticker =  tick(StockSymbol,P,T),
    sendMsg(CID,OA,answer,monitor(LuxuryCarManufacturer, Ticker)) .
```

An organizational agent (OA) typically manages several personal agents and delegates queries of external agents to their personalized views. It might e.g. be used to implement privacy and security policies. For instance, the following rule detects suspicious logins by assessing the IP numbers of the login events from the same user login.

```
rcvMsg(XID,Protocol,From,request,login(User,IP)) :-
  % if the next follow up event (@count(1)) that follows the
  %previous received login is send from another IP (IP2!=IP)
  @group(g1) @count(1)
  rcvMsg(XID,Protocol,From,request,login(User,IP2))  [IP2!=IP],
  println(["Suspicious login",User,IP,IP2]," ").
```

4.3.5 Conclusion

Our rule-based multi-agent approach for implementing personalized views benefits from the declarative properties of rule programming, the expressiveness of the used rule language, and the underlying formal logic semantics. A declarative rule-based method for integrating knowledge into adaptable and adaptive personalized views provides higher levels of flexibility to describe in a semantic way the multiple aspects and contextual dimensions which personalized views might take. In particular, the agent's personalized views might be enriched by the pragmatic context in which they are used, such as communicative situations, purposes or individual goals and values, organizational norms, legal norms, etc.

Chapter 5

Conclusion and Outlook

In this report we described the validation and evaluation of our prototypes in the project *Corporate Semantic Web* during the last milestone phase. We presented several prototypical implementations as a proof of our conceptual Corporate Semantic Web architecture presented in the last report, covering the three pillars Corporate Ontology Engineering, Corporate Semantic Collaboration, and Corporate Semantic Search. The prototypes were partially developed and evaluated in tight co-operation with our industrial partners.

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
WP 3 Task 3.4	Validation of results	03/12-01/13
WP 3 Task 3.5	Evaluation of results	03/12-01/13

Work package 4	**Search contextualization**	02/11-01/13
WP 4 Task 4.4	Validation of results	03/12-01/13
WP 4 Task 4.5	Evaluation of results	03/12-01/13

Work package 5	**Knowledge Extraction from User Activites**	01/08-04/12
WP 5 Task 5.4	Evaluation of the prototypical implementation	02/11-04/12

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
WP 8 Task 8.4	Validation	03/12-08/12
WP 8 Task 8.5	Evaluation	08/12-01/13

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
WP 8 Task 8.3	Prototypical Implementation	08/11-02/12
WP 8 Task 8.4	Validation	03/12-08/12
WP 8 Task 8.5	Evaluation	08/12-01/13

Work package 11	**Ontology cost models for enterprises**	02/11-01/13
WP 11 Task 11.4	Validation of the cost model in an industrial scenario	02/12-07/12
WP 11 Task 11.5	Final evaluation	08/12-01/13

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
WP 12 Task 12.4	Validation	02/12-08/12
WP 12 Task 12.5	Evaluation	08/12-01/13

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
WP 14 Task 14.3	Integration of personalized views with enterprise knowledge	12/11-1/13
WP 14 Task 14.4	Evaluation	12/11-01/13

Appendix B

Acknowledgment

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] Costin Badica, Lars Braubach, and Adrian Paschke. Rule-based distributed and agent systems. In *RuleML Europe*, pages 3–28, 2011.

[2] 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.

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

[4] 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.

[5] Harold Boley, Adrian Paschke, and Omair Shafiq. Ruleml 1.0: The overarching specification of web rules. In *RuleML*, pages 162–178, 2010.

[6] Booktate. Booktate website. http://booktate.com/, 2012.

[7] Lars Braubach, Alexander Pokahr, and Adrian Paschke. Using rule-based concepts as foundation for higher-level agent architectures. In Kuldar Traveter Adrian Giurca, Dragan Gasevic, editor, *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, pages 493–524. Information Science Publishing, 5 2009.

[8] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52, 1998.

[9] Robin Burke. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*, volume 69, 2000.

[10] Robin Burke. The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.

[11] S. Chakravarthy and D. Mishra. Snoop: an expressive event specification language for active databases. *Data Knowl. Eng.*, 14:1–26, November 1994.

[12] H. Cunningham. Gate: general architecture for text engineering. http://gate.ac.uk.

[13] et al. Dietmar Jannach. *Recommender Systems: An Introduction.* Cambridge University Press, 2011.

[14] Diigo Inc. diigo website. http://diigo.com, 2012.

[15] faviki. faviki - tags that make sense. http://www.faviki.com.

[16] Peter Haase, Daniel Herzig, Mark Musen, and Thanh Tran. Semantic wiki search. In *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece*, pages 445–460, 2009.

[17] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[18] Annika Hinze, Ralf Heese, Markus Luczak-Rösch, and Adrian Paschke. Semantic enrichment by non-experts: Usability of manual annotation tools. In *International Semantic Web Conference (1)*, pages 165–181, 2012.

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

[20] Ping-Ju Hsieh. Administration service for the tourist information system. Master's thesis, Computer Science Department, The University of Waikato, June 2008.

[21] 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.

[22] Qi Lin. Semantic multimedia presentation tool, 2012.

[23] Olive Tree Bible Software, Inc. Bible+. available online at http://itunes.apple.com/, 2012.

[24] A. Paschke. Eca-lp / eca-ruleml: A homogeneous event-condition-action logic programming language. In *RuleML-2006*, Athens, Georgia, USA, 2006.

[25] Adrian Paschke. Eca-ruleml: An approach combining eca rules with temporal interval-based kr event/action logics and transactional update logics. *CoRR*, abs/cs/0610167, 2006.

[26] Adrian Paschke. Rules and logic programming for the web. In *Reasoning Web*, pages 326–381, 2011.

[27] Adrian Paschke. A semantic rule and event driven approach for agile decision-centric business process management - (invited paper). In *ServiceWave*, pages 254–267, 2011.

[28] Adrian Paschke and Rehab Alnemr. The Rule Responder Distributed Reputation Management System for the Semantic Web . In *Proceedings of the 4th International Rule Challenge.* CEUR. Vol-649, 2010.

[29] Adrian Paschke and Martin Bichler. Knowledge representation concepts for automated sla management. *Decis. Support Syst.*, 46(1):187–205, 2008.

[30] Adrian Paschke and Harold Boley. Rule markup languages and semantic web rule languages. In Kuldar Traveter Adrian Giurca, Dragan Gasevic, editor, *Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches*, pages 1–24. Information Science Publishing, 5 2009.

[31] Adrian Paschke and Harold Boley. Rule responder: Rule-based agents for the semantic-pragmatic web. *International Journal on Artificial Intelligence Tools*, 20(6):1043–1081, 2011.

[32] Adrian Paschke, Harold Boley, Zhili Zhao, Kia Teymourian, and Tara Athan. Reaction ruleml 1.0: Standardized semantic reaction rules. In *RuleML*, pages 100–119, 2012.

[33] 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.

[34] Adrian Paschke, Gökhan Coskun, Ralf Heese, Radoslaw Oldakowski, Mario Rothe, Ralph Schäfermeier, Olga Streibel, Kia Teymourian, and Alexandru Todor. Corporate semantic web report v: Prototypical implementations and working packages in project phase ii. Technical Report TR-B-12-04, Freie Universität Berlin, 2012.

[35] 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.

[36] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based recommendation systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[37] E. Peis, J. M. Morales del Castillo, and J. A. Delgado-Lopez. Semantic recommender systems. analysis of the state of the topic. *Hipertext.net*, 6:(online), 2008. Author: E. Peis; J. M. Morales-del-Castillo; J. A. Delgado-Lopez Location in hierarchy: Home -¿ Number 6, May 2008 -¿ Recommender URL of this page: http://www.hipertext.net/english/pag1031.htm Citation: E. Peis; J. M. Morales-del-Castillo; J. A. Delgado-Lopez. Semantic Recommender Systems. Analysis of the state of the topic [on line]. "Hipertext.net", num. 6, 2008. ¡http://www.hipertext.net¿. ISSN 1695-5498.

[38] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The adaptive web*, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[39] Alexa Schlegel, Ralf Heese, and Annika Hinze. Visualisation of semantic enrichment. In *Workshop on Interaktion und Visualisierung im Daten-Web*, September 2012.

[40] 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.

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

[42] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[43] 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 Las Vegas,US*, pages 00–00. IEEE Computer Society, 2010.

[44] Mari Carmen Suárez-Figueroa and Asunción Gómez-Pérez. Ontology Requirements Specification. In Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi, editors, *Ontology Engineering in a Networked World*, chapter 5, pages 93–106. Springer Berlin Heidelberg, 2012 edition, March 2012.

[45] Kia Teymourian, Gökhan Coskun, and Adrian Paschke. Modular upper-level ontologies for semantic complex event processing. In *WoMO*, pages 81–93, 2010.

[46] Kia Teymourian and Adrian Paschke. Semantic rule-based complex event processing. In *RuleML 2009: Proceedings of the International RuleML Symposium on Rule Interchange and Applications*, 2009.

[47] Kia Teymourian and Adrian Paschke. Towards semantic event processing. In *DEBS '09: Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*, pages 1–2, New York, NY, USA, 2009. ACM.

[48] Kia Teymourian, Malte Rohde, and Adrian Paschke. Fusion of background knowledge and streams of events. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, DEBS '12, pages 302–313, New York, NY, USA, 2012. ACM.

[49] Thomson Reuters Inc. Open Calais website. http://www.opencalais.com/.

[50] veeeb – Top 21 GmbH. veeeb website. http://www.veeeb.com, 2012.

[51] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Commun. ACM*, 51(8):58–67, 2008.

[52] Hans Weigand and Adrian Paschke. The pragmatic web: Putting rules in context. In *RuleML*, pages 182–192, 2012.

[53] Zemanta Ltd. Zemanta website. http://www.zemanta.com/.

[54] Zhili Zhao and Adrian Paschke. Event-Driven Scientific Workflow Execution. *6th International Workshop on Event Driven Business Process Management (edBPM 2012)*, September 2012.

[55] Zhili Zhao, Adrian Paschke, Chaudhry Usman Ali, and Harold Boley. Principles of the symposiumplanner instantiations of rule responder. In *RuleML America*, pages 97–111, 2011.