A Semantic Web Approach to CE

Abstract

Collaboration and communication is vital for concurrent engineering and concurrent enterprises. Studies show that members of product development teams spend more and more time for communication, information and document management and have less and less time for the creative part. Online and off-line collaboration tools address the needs of CE but searching for information and sharing knowledge still follows the old metaphor of knowledge management, i.e. knowledge can only be shared (and accessed) when a common terminology is used. To overcome this limitation, today's search engines are starting to incorporate lexical mapping schemes but those are not adapted to the specifics of the engineering process.

Here the emerging Semantic Web technology offers new possibilities through modelling and using a domain ontology (domain knowledge) to better support users of different disciplines to exchange information and knowledge without using the other parties terminology. This paper presents one of the first approaches to tackle information and knowledge sharing by using the latest - still emerging – Semantic Web technologies. We present the system's motivation, architecture and use examples. Furthermore we report on results from user test (done by users from Schenck Pegasus GmbH and ItalDesign Giugiaro). We describe the experience that we gathered with the different Semantic Web technologies and give an outlook to future research and development needs.

Keywords

Knowledge management, information retrieval, heterogeneous information integration, collaboration support, concurrent engineering, Semantic Web technology

Introduction

Concurrent Enterprising and Concurrent Engineering heavily requires online and offline communication. In the past years many methodologies, technologies and tools have been developed to support engineering processes in concurrent environments, e.g. video conferencing, collaborative CAD environments, PDM/PLM systems, etc. These approaches are still lacking a consistent and integrated management of information and knowledge. The users have to know which information is where and how to access it. For successful retrieval, queries have to be formulated in the right syntax and structure, and so on and so forth.

To newly tackle these well known problems, we took a Semantic Web Technology based approach. First of all, one has to ask what it means, to support engineering processes or more generally innovation processes. In our opinion it consists of supporting the user

- in finding information of various kinds and formats,
- in helping to aggregate information to knowledge,
- in finding experts from other disciplines to discuss new ideas or problem with the current designs.

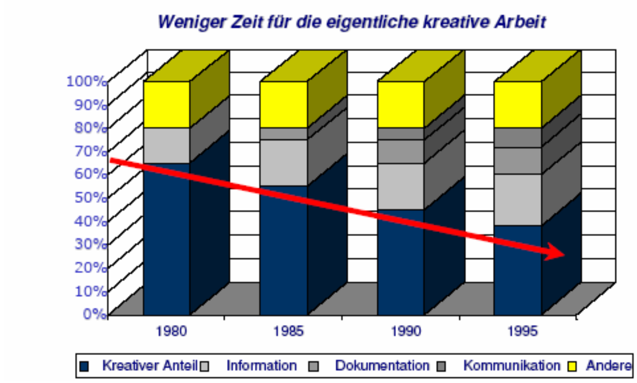
All that shall be possible from one interface in an easy-to-use manner without forcing the user to use a special terminology or a concrete query language. Here is where Semantic Web Technology comes into play. Heterogeneous information integration based on Semantic Web Technology such as ontologies, agents, etc. for improved off-line collaboration (information and knowledge retrieval and exchange) as well as for improved online collaboration in multi-disciplinary team has been explored within the EC project WIDE and will be described in this paper.

The WIDE project and the corresponding system is one of the first attempts to evaluate rapidly developing Semantic Web Technologies within in industrial environment which consists of product designing and engineering. Two companies, ItalDesign Giugiaro in the design stage and Schenck Pegasus in the engineering and testing stage, serve as industrial partners specifying their needs and evaluating the system. The WIDE system has already undergone two evaluation phases heavily influencing the design and development of the system.

This paper will discuss the information and knowledge retrieval and exchange needs for engineering processes also covering online collaboration in a new flavour, the WIDE systems architecture and functionality, the findings from user tests and related to Semantic Web Technology and finally the implications for further developments especially in terms of Semantic Web Technology for improved Concurrent Engineering and Enterprising.

Current situation

Over the last 25 years the portion of creative work in the engineering process was constantly decreasing as statistical data shows (Eigner/Knoche). This effect is due to shortened development cycles, more interaction within different teams in the development process either belonging to the same or different companies.



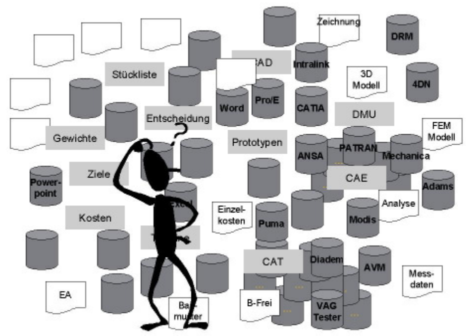


Fig 1. - left: ratio between managing information and documentation and creative work (Eigner/Knoche)

Fig 2. - right: typical data source chaos in a CAT environment, CAT= Computer Aided Testing

Since the trend from a sequential organisation to a concurrent/simultaneous organisation of the development process that entails increased communication needs is irreversible, better information/knowledge retrieval and communication tools are needed. Current tools have certainly improved the situation over the last couple of years, but still the user needs to know where to search, how to search, how to express his search query so that the system can interpret it, etc. (see figure 2). Search engines such as Google are still much to pre-mature to provide good results for complex queries.

Presentations from car makers during the VDI (Verein Deutscher Ingenieure) -Congress "Testing and Simulation: Measurement and Trial Technology" (April 2003) about their internal efforts in the area of information inquiry highlight this. They still use systems where the search function uses pure text-based pattern matching on meta-data without any additional intelligence (e.g. synonym mapping), or where the user must manually navigate through file structure trees. They also stressed the importance of information retrieval from worldwide distributed sources to save time in the development process. Some of them tackle the problems with different terminology/languages, different structures and syntaxes by forcing to use one information backbone, unique identifiers, etc. This might work well for information accessible via the product structure ID but efforts are high and still limited to product data and corresponding management systems. Styling and design, engineering and testing departments are only starting to be incorporated.

Many car makers (industries in general) are looking for systems that support semantic search, finding relevant information in different sources using a 'human-friendly' query language. Relevant information also comprises information that is related and linked to an information already found. But the information needs are not limited to just search for information, they also comprise finding experts for collaborative problem solving, looking for new material and production technology, etc.

These observations motivated us to do some research and development aiming at improved information retrieval and knowledge exchange for online and off-line concurrent engineering processes. For this end, we explored (used and further developed) Semantic Web Technology (see below).

Related Work

SEWASIE (Semantic Webs and Agents in Integrated Economies) [16] is an European IST project aiming at the creation of a semantic search infrastructure and at investigating the applicability of Semantic Web and Agents technologies to the context of integrated economies. In such field, it seeks methodologies and tools to provide integrated and sophisticated services to retrieve, share and communicate information in a multi-enterprise context.

The first step in SEWASIE is to semantically enrich the information sources, by means of ontologies and intelligent agents, so to build-up a network of distributed intelligent nodes. The second step consists in the realization of a component to manage queries submitted by users and target them to the proper sources, combine results and display them to the user in an integrated way. Agent technology is used to contact the sources on the network and bring back the results produced. Network nodes are independent entities, which semantically enrich the data sources by connecting the information they store to the domain ontologies and other metadata. Query results are eventually customized and displayed according to users' preferences. This project is quite similar to WIDE in the technologies adopted and in the objective pursued. Differences are about doing any online collaboration and coping for different user types and tasks. Though SEWASIE is not thought for a specific domain, its ontologies must be targeted to some particular need, otherwise they would get too general to reach a good-enough level of precision.

The SWIM (Semantic Web Integration Middleware) system [15] addresses the problem of the integration of data in the Semantic Web, by managing the links between existing information sources and ontologies. This system is especially focused on the problem of mapping (i.e. rewriting) the queries expressed in a specific language (e.g. RQL) to the specific structure and languages adopted by the various information sources available (e.g. XML, SQL, etc.). In other words it can create RDF views of other lower level data formats. The mappings are realized by means of Datalog-like rules. A query rewriting approach like that present in SWIM is adopted in the WIDE system too, where all the transformations of data formats and syntaxes are to be made in a way as transparent from the information sources as possible, not to be too invasive and pose any constraints. However, there are no relationships to other features and requirements addressed by WIDE, like the absence of any need for a common language among the users submit queries to the system. The focus of SWIM is more targeted at the integration of raw data stored at the various sources, rather than providing a full-fledged platform considering all the aspects of the user interaction. Yet it shares with WIDE some of the core features at the basis of the mapping algorithms.

Research Approach

To tackle the described information management and knowledge sharing problem, a Semantic Web Technology based approach has been chosen.

The Semantic Web, firstly introduced by Berners-Lee [1], aims at evolving the Internet from a human-readable platform for information exchange into a machine-understandable medium where semantically described information sources and services can be automatically combined to fulfil certain tasks defined by a user. For this end, technologies and tools are under development and rapidly evolving which seam to be well suited for the problem we are facing with today's heterogeneous information structures. Among those tools and technologies there are:

- Protégé [8] for modelling ontologies
- RACER [10] for doing reasoning on OWL ontologies
- Sesame [9] for querying and navigating RDF sources
- RDF(S) [5, 6] for modelling an RDF source and describing its content
- RQL [3] for querying RDF sources
- OWL [7] for describing ontologies
- W3C result format [12] for transmitting RDF results

These tools are used to develop, query and reason about ontologies. Ontologies are semantic conceptions of real world domains. Ontologies typically comprise taxonomies and thesauri which allow for classifying terms, to contextualize terms and to replace terms by synonyms.

Query languages such as RQL (the equivalent of SQL for RDF sources) provide the possibility to query information sources modelled with RDF(S). Tools such as Sesame (an RDF suite) allow to query and reason about RDF sources [13]. Higher level inference techniques, e.g. transitivity of relationships [14], hierarchies of concepts and relationships, can be incorporated to provide more semantic query facilities than traditional relational databases.

Nonetheless, the query has to match the structure of the information source as with other query languages. Since various heterogeneous information sources are different in structure and might not all be modelled with RDF(S), there is a need for terminological and structural query mapping and rewriting. We are performing different kinds of rewriting on the various levels of our system. Starting with user queries which are interpreted using a domain ontology up to mapping the so called system queries to different query languages syntactically and structurally depending on the information source.

One of the basic principles of our approach is motivated by the fact that different users (different disciplines) use different words (terms) when looking for the same or similar kind of information. In contrast to other approaches we do not want to force the users to use one terminology, instead we are mapping the user's terminology to reference terms defined in the domain ontology. This renders the idea of knowledge exchange to be possible only when same terms are used more flexible, natural and practical.

Usage scenarios

When designing a system for knowledge sharing and collaboration support, one should think about why and how users look and search for information and in which situation they might be interested in online collaboration.

From our point of view, each information retrieval session is motivated by a task, e.g. the development of a new product or more likely a sub-task of such a general task. Each information retrieval session has an aim, e.g. what kind of new materials are there to create a part, what kind of new production technologies are around to produce the part, etc.

We believe that current search technology is inadequate for professional search tasks because it is context-free. Therefore we introduced – amongst other things, e.g. interactive query development (see below) - the workbook concept. The workbook concept is a small editor on the user interface that allows the user to enter a task description, that automatically keeps track of queries and results that allow the user to comment on those results and to delete queries that yield irrelevant results. Thus, the workbook acts as an editable session recorder covering the actions of the user, his intend and allows him to document what he did and why and which results have been produced. The workbook content can be exported and used in other applications, e.g. to view the collection of results (see the below ASAM-ODS example).

During this process of information / knowledge retrieval, the need to collaborate can rise. Possible collaboration scenarios might be, e.g.:

- Looking for another opinion: user A found something that he likes to discuss with somebody else, e.g. I found that bit here, what do you think?
- Looking for help: user A needs somebody that helps him along with his task, e.g. I have found this problem here with prototype X, do you have an idea how to solve it? Can you refer to some documents where it is described?
- Etc.

With a problem-, task,- or problem solving-oriented ontology and corresponding content some of those events could be supported (some of the traditional knowledge management approaches focus here) but we believe that human-to-human collaboration might still be needed in many cases.

To address those cases, we introduced a collaborative component into our system which allows to consult another user. When he accepts the request, queries and results as well as their concepts and relationships are shown to the partner. By using terminology mapping for both involved users, the system helps them to understand the other person's terms by showing how each user's term relate to the reference domain ontology's terminology.

System architecture

The requirements imposed by the goal to support different users aiming to search and share information and knowledge in a heterogeneous environment are addressed by a system architecture as described in the following.

The building blocks of the architecture are: the User Interface, the Meta Level, the Agency and the Content Level. The Agency - a multi agent system - is used to "glue" those components together, as shown in Figure 3.

We have established links to the following information sources:

- an internal RDFS source acting as a semantic information source where we experiment with services future semantic information sources should provide – this source contains information and meta data about documents relevant for different stages in the design of cars (provided by ItalDesign Giugiaro and Schenck)
- an ASAM-ODS source containing test results from an engineering department (provided by Schenck Pegasus GmbH)
- the internet, esp. search results found via the Google API

The RDFS source is addressed using RQL. The ASAM-ODS data base has its own query language, so that standard RQL queries have to be mapped to be able to be applied to it; the same holds true for the Google API.

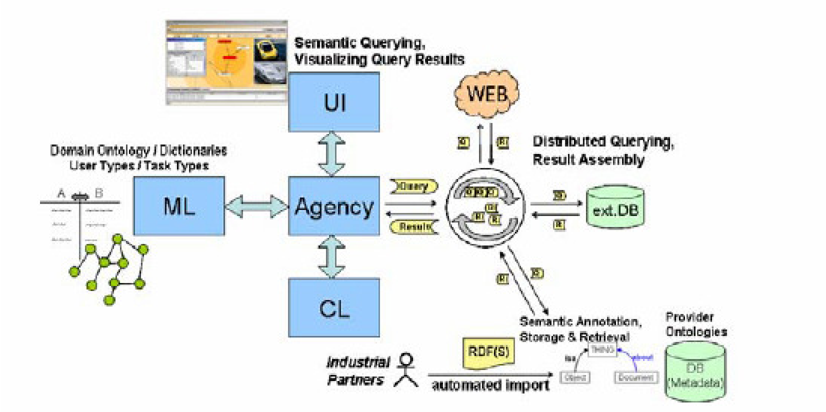


Fig 3. - The WIDE system architecture

The User Interface

The User Interface (UI) provides a graphic front end to the user and supports the incremental development of user queries in an alphanumeric or graphic way. By easy to use drag and drop operations the user can successively build up his query. This interactive and incremental query development process is supported by the Meta Level and the domain knowledge contained in the Meta Level and brought into BNF form [2]. Furthermore, the UI presents the returned results and its relationships (semantics) in a graph-based structure. This graph structure can be navigated by the user in order to explore the returned results and its metadata. Based on the returned graph structure and the corresponding metadata the current user query can be refined or a new one can be developed

The workbook concept in the user interface supports the user in documenting and structuring his session. He/she can comment on the aim of the session and the results.

Concepts for online collaboration: A user working with the system, e.g. to retrieve knowledge about a certain process or looking for information/knowledge can contact other users logged-in. The local user interface is partly replicated on the remote side. References to the domain ontology (the internal terminology) help the users to understand the terminology used by the partner. As far as we know, this is a new kind of trying to bridge the gap between different terminology uses in collaborative sessions.

The Meta Level

The Meta Level (ML) supports the user query development. The main purpose of the ML is the semantic processing of user queries into system queries and the semantic processing of the returned results. To do this, the ML uses a domain ontology (for car design and engineering), together with a Task Type ontology (knowledge about the different tasks carried out in the domain) with almost 800 concepts, User Type ontology (knowledge about the profile of the different users in the domain), and dictionaries of description terms and User Type terms. All these different and interrelated kinds of knowledge are used to produce the system queries that are then passed to the Agency. The returned results undergo a similar semantic processing as the user queries. They are semantically processed in order to associate them with the appropriate concepts in the domain ontology and finally display them in a meaningful graph structure by the UI.

The Agency

The Agency subsystem identifies and locates information sources in the Content Level to which the system queries can be sent to produce effective returns. The Agency also provides the system's gateway to the Web, which is also considered part of the Content Level. Essentially, Web sites and Web search engines are treated as weakly structured information sources. Besides the planning and execution of queries, the Agency's responsibilities are also to collect and to transform the results of the heterogeneous information sources into a common result format on which the ML is performing semantic processing and reasoning.

The Content Level

The Content Level (CL) consists of different information sources (RDF sources [5, 6], ASAM/ODS [4], relational databases and the web) that vary in their semantic richness. Since those information sources might have a different structure than the ML domain ontology, one of the tasks of the CL is to adapt the system queries to the query language understood by the information sources (syntactically, terminologically, and structurally). The purpose of the CL is not only to retrieve results for a given query but also to return semantic context that helps the ML to reason about the result and is used for visualising the semantics by the UI.

One of the main concepts and principles of the WIDE system is that the domain ontology and the information sources are modelled independently. The semantically richer an information source is, the easier it is to establish a relationship with the domain ontology through the diverse forms of mapping. For semantic information sources we propose the following functionality:

- publish its structure
- support synonyms for the terms contained (for improved matching probability)
- support mapping and rewriting facilities for incoming queries
- provide a semantically rich result format from which other components can derive information for result ranking, filtering and display
- implement semantic search algorithms

Use example for the WIDE system in an engineering/testing department

Mr. Power (an engineer and expert for gasoline injection systems) gets the task to increase the power of an engine with 2 litres cubic capacity and 85 kW. The performance of this engine shall be increased by 20 %. Furthermore, the engine emissions must comply with emission standard EURO 4, when the engine is used in the car “Bobby Car II”.

Mr. Power logs into the WIDE system (WIDE knows from the user model of Mr. Power that he is an engineer). He selects the process ‘increase engine power’ known to the WIDE domain ontology.

The engineer searches for engines with 2 litres cubic capacity and 85kw.

He interactively developed the following user query (UQ) using WIDE’s drag and drop functionality (see figure 4)

UQ_1: engines with (cc = 2.0l AND PowerMax = 85kw)

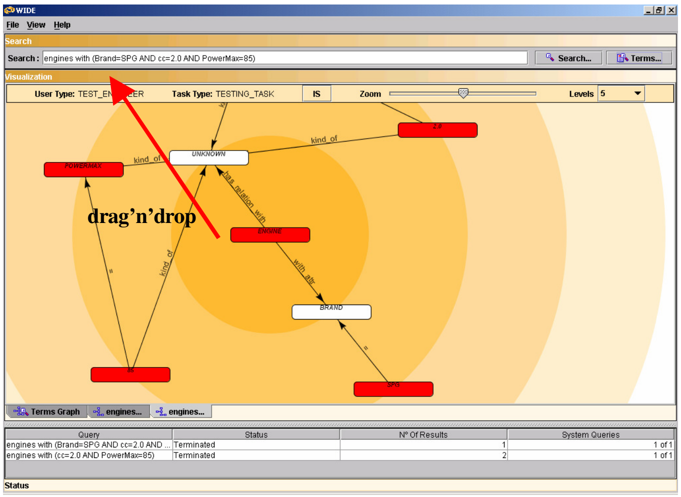
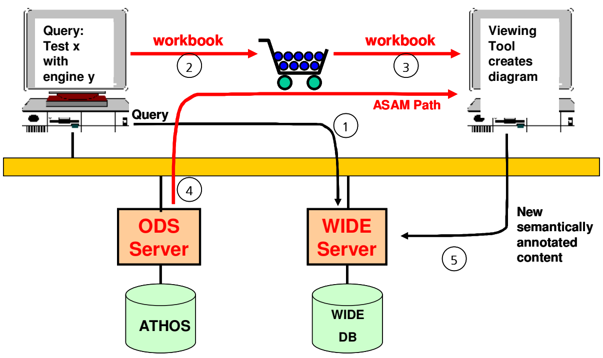


Fig 4. - Interactive query development per drag and drop

WIDE retrieves one engine from an ASAM-ODS information source. The user inspects the meta data of that engine and finds its SerialNumber. To check whether it is suited as a base engine for the power increase, he is interested in full load tests. He types in

UQ_2: Documents about tests with (description = “Full Load Test” AND engine with SerialNumber=S0025-014

WIDE retrieves the meta data of the test and the corresponding URL of the test results (vector in the ASAM-ODS data base). To visualize the test results the user launches an appropriate tool. To pass the data to the tool, he exports the workbook (see figure 5) which contains the steps taken and the research result (URL, etc.)..



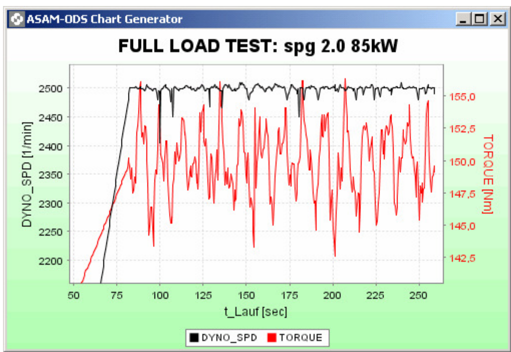


Fig 5. - left: the workbook concept

Fig 6. - right: diagram showing oscillations

With this tool, he is able to visualize the data contained in the referenced test. The diagram shows an oscillating strange behaviour.

As Mr. Power does not know how to interpret the diagram and to compensate the oscillations, he contacts an expert in mechanical vibrations (Mr. Vibration). WIDE sends the information to the remote expert. Mr. Vibration explains the problem to Mr. Power. The engine produced rotational oscillations at certain load points during the test (at a speed around 2500 rpm). To use the engine for the power increase task Mr. Vibration recommends to change the bearing rods.

For future reference, both decide to store the diagram in the WIDE internal information source (see step 5 in figure 5), adding the cause of the oscillation as semantic annotation, thus creating new content and knowledge in the collaborative session.

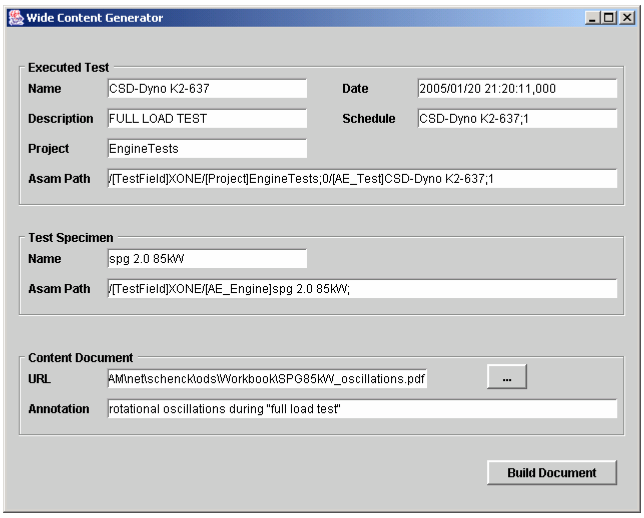


Fig 7. - WIDE content generator allow to semantically annotate the diagram and store it in WIDE

Findings

During the two cycles of user tests we collected user feedback that drove the further developments. To summarize, the user liked

- the interface as a unified means to search internal and external information sources
- the way they can create and modify queries either by typing or by dragging and dropping terms from the graphics window to the query window
- the intuitiveness and expressiveness of queries: they can formulate complex queries in quasi-NL (natural language) asking for relationships of entities even distributed over different information sources, e.g.
  - test results of emission tests for cars with 8 cylinder engines
  - pictures of aggressive cars
  - etc.
- the way how results are visualized (graphically connected according to their semantic relations)
- that they can ask the WIDE domain ontology about what terms and processes it knows and that they can further explore/navigate this information space before actually entering a query
- that they can establish collaboration via the WIDE system
- the workbook approach to structure and document their work plus the possibility to export results to a 3rd party tool

The main wish that is still left open from the users' point of view, is the integration of more information sources into the semantic approach in shorter time. This yield to a major issue under discussion today in the Semantic Web community: faster automatic knowledge creation which will be certainly part of the research community for the next couple of years.

<u>Advantages and problems encountered with emerging Semantic Web technology</u>

The main advantage of Semantic Web technology such as RDF and OWL is the ability to describe content in a more semantic manner as with standard relational techniques. In addition to the mere meta data as known from relational databases, one adds hierarchies of concepts and relationships thereby adding meaning to the information stored. This additional information (semantics, meta data) can be used for inference to increase the quality and quantity of search results.

For example, if somebody looks for documents about doors, patents, customer requirements, etc. are returned because they are all known to by types of documents to the semantic information source. If one looks for patents, the information that patent is understood as a type of document is also returned for further filtering and ranking the results.

By using taxonomies, queries can be formulated less formally and users still retrieve relevant results. We see the incorporation of semantics and the support of taxonomies as a main advantage, providing the user with more intuitive and still complex query formulation. Certainly, query mapping can also be done with traditional query languages such as SQl, but Semantic Web technology has more intrinsic support for it.

The problems that we encountered with emerging Semantic Web technology can be summarized as follows:

- Ontology building tools (e.g. Protegé) need a lot of ontology modelling know-how. Ontology engineering [11] is still not perfectly understood, esp. in complex environments such as product designing and engineering
  - Changes in the structure need a re-entry of the content
- RQL seam to be an appropriate query language for RDF resources. However, the performance of RDF sources when answering RQL queries (e.g. Sesamé) heavily depends on the structure of the RQL query
  - We are performing RQL query normalization for improving query processing performance, e.g. tens of minutes vs a couple of seconds for the normalized form
- Expressiveness of ontology modelling languages
  - Some real world relationships e.g. instances that refer to concept cannot be modelled in a straight-forward manner but need work-arounds (e.g. $document_a$ about doors, $document_b$ about doors – document a and b are instances of documents whereas doors is a concept).
  - Concepts need to have unique names, but what if SEAT (the car brand) and 'seat' shall be part of one ontology? Again work-arounds are needed.
- Limitations of reasoners: hierarchies of transitive relationships cannot be used in a straight-forward manner.
  - Imagine a relationship 'has'. 'Has' might bind 'Engine' and 'Cylinder' in the way: 'Engine has cylinder'. The relationship with the name 'has' cannot be used between two other concepts, e.g. 'Cylinder has valves'. With a transitive 'has', one could easily infer that engine has valves. Instead a different relationship needs to be introduced, e.g. 'another_has'. This new relationship might be inherited from 'has' being a transitive relationship as well, resulting in: 'Engine has cylinder another_has valve'. A naïve approach would be to think that reasoners would be able to infer that 'Engine has valve' due to the fact that both 'has' and 'another_has' are both transitive and 'another_has' is derived from 'has'. Unfortunately, the reasoners we have used (the one from Sesame and RACER) are not behaving like this instead they follow the strict 'description logic' where a relation is different if is connects different concepts. Again work-arounds are required, e.g. the pre-procesing of the relationships as done in the Meta Level.
- Reasoners for OWL are still pre-mature. Most of them do not support the inference larger ontologies in a usable way.
- Even when using RDF and RQL, query mapping and rewriting is needed because Sesame just uses the rules of description logic plus the user-defined ones when trying to retrieve the results for a query. Mapping, rewriting and heuristic reasoning can enhance result quality and quantity.
- Lack of standard Ontologies, esp. for specific domains are neither defined nor accessible on the Web
- Missing standard for formatting and transmitting RDF results
  There is a RDF results format proposal by W3C but it leaves open a lot of flexibility with respect of what is contained in an RDF result stream. From our experience it is not sufficient to just return the RDF fragments found by a RQL query. Instead we semantically extend the results by their context, e.g. information about parent concepts, siblings, etc. and information about the mapping that was done in the querying process (so-called 'return-for' information). This extended RDF results format better supports reasoning and ranking of the results. A standard RDF results format shall not only be a syntax formulation but also require that certain semantic information are contained.

Conclusions

We have presented an information retrieval and knowledge sharing system for concurrent engineering and enterprises. The system supports offline and online collaboration. For this end, it is using newest and still developing Semantic Web technologies, such as OWL. Compared to the more traditional information retrieval approaches it has the clear benefit of incorporating more semantics into the search process for better result definition and quality. The main benefit we see on the users side is how he is supported by a knowledgeable system in the query generation process based on ontologies. The use of ontologies and taxanomies (thesauri and synonyms) relieves him from using a certain terminology thus improving the access to knowledge provided by other users in different terminology.

The main conclusion of our work is that the vision of the Semantic Web and its technologies can significantly improve the information and knowledge retrieval and sharing. Still it is not completely understood what the different components in the Semantic Web are and what functionality they have to provide to bring the Semantic Web to reality. By exploring Semantic Web technology for information and knowledge retrieval and sharing, we gathered a lot of experience and derived suggestions for components' functionalities and architectures.

The system has undergone two cycles of user testing which gave many hints for further developments. The main topics for future work from the technology point of view are:

- Easy-to-use content annotation and knowledge creation tools for end-users
- Better expressiveness of ontology modelling tools
- Decreasing knowledge maintenance efforts through better knowledge/information acquisition tools
- Better information mining tools to get meta data out of documents of different types more automatically
- Easy to configure wrapping and mapping tools to legal systems, e.g. PDM/PLM systems

Acknowledgements

This work was funded in part by the European Commission Grant #IST-2001-34417, Semantic Web-based Information Management and Knowledge Sharing for Innovative Product Design and Engineering (the WIDE project).

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific Ameri-can*, pp 34–43, May 2001.

[2] Backus-Naur form (BNF), WIKIPEDIA, <http://en.wikipedia.org/wiki/Backus-Naur Form>.

[3] The RDF Query Langauge (RQL), FORTH Institute of Computer Science, <http://athena.ics.forth.gr:9090/RDF/RQL/>.

[4] Association for Standarisation of Automation and Measuring Syetems (ASAM) Open Data Service (ODS), <http://www.asam.net/01 asam-ev 01.php>.

[5] Resource Description Framework (RDF), W3C Semantic Web Activity, Technology and Society Domain, <http://www.w3.org/RDF/>.

[6] RDF Vocabulary Description Langauge 1.0: RDF Scheme, W3C Technical Reports and Publications, <http://www.w3.org/TR/rdf-schema/>.

[7] Sean Bechhofer, Phillip Lord, and Raphael Volz. Cooking the Semantic Web with the OWL API. In *2nd International Semantic Web Conference, ISWC*, volume 2870 of *Lecture Notes in Computer Science*, Sanibel Island, Florida, October 2003. Springer.

[8] The Protégé Ontology Editor, Stanford Medical Informatics, Stanford University School of Medicine, <http://protege.stanford.edu/>.

[9] J. Broekstra and A. Kampman and F. van Harmelen, "Sesame: A generic architecture for storing and querying RDF and RDF Schema, International Semantic Web Conference (ISWC), pp 54-68, 2002.

[10] RACER: Semantic Middleware for Industrial Projects based on RDF/OWL, <http://www.cs.concordia.ca/˜haarslev/racer/>.

[11] A. Gomez-Perez, M. Fernandez-Lopez and O. Corcho, "Ontological Engineering," London:Springer-Verlag, 2004.

[12] A. Seaborne, Recording Query Results, W3C Discussion document http://www.w3.org/2003/03/rdfqr-tests/recording-query results.html

[13] RDF semantics W3C Recommendation 10 February 2004 http://www.w3.org/TR/rdf-mt/

[14] S. Staab, M. Erdmann, and A. Maedche. An Extensible Approach for Modeling Ontologies in RDF(S). In First ECDL'2000 SemanticWebWorkshop, Lisbon, Portugal, 2000.

[15] V. Christophides, G. Karvounarakis, I. Koffina, G. Kokkinidis, A. Magkanaraki, D. Plexousakis, G. Serfiotis, and V. Tannen. The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware. First International Workshop on Semantic Web and Databases, co-located with VLDB 2003, Humboldt-Universitat, Berlin, Germany, September 7-8, 2003.

[16] SEWASIE Project Web site: http://www.sewasie.org.