Semantic Technologies for Enterprises

Frithjof Dau

SAP Research Dresden

Abstract. After being mainly a research topic, semantic technologies (ST) have reached an inflection point in the market. This paper discusses the benefits (data integration and federation, agile schema development, semantic and collaborative / social computing search capabilities) and costs (namely technical, modeling, measuring and educational challenges) of Semantic Technologies with respect to their utilization in enterprises.

Keywords: Semantic Technologies, Semantic Web, Enterprise Applications

1 Introduction

In its Spring 2009 Technology Forecast [1], PwC (PricewaterhouseCoopers) predicts that "during the next three to five years, we [PwC] forecast a transformation of the enterprise data management function driven by explicit engagement with data semantics." A recent (spring 2010) report [2] in which 50 high-level decision makers have been interviewed states that "the next generation of IT will be structured around unified information management, Enterprise-level, semantically aware search capabilities, and intelligent collaboration environments - all delivered through dynamic, personalized interfaces that are aware of context". Taking these two quotations and their sources into account, there is a clear indication that after being mainly a research topic in Academia, now semantic technologies (ST) have reached an inflection point in the market. This paper will discuss the pros and cons of ST with respect to their utilization in enterprises.

The herein presented discussions and insights particularly stem from the author's experience in the research project Aletheia, lead by SAP, where he has been responsible for the ST layer of Aletheia. The next paragraph summarizes Aletheia and is taken from an SAP-internal whitepaper [3].

The Aletheia research project investigated how current semantic technologies can be applied in enterprise environments to semantically integrate information from heterogeneous data sources and provide unified information access to end users. Often, related product information is spread across different **heterogeneous data sources** (e.g. product information in a database is related to a PDF manual for that product or an entry on the producing company in a spreadsheet). **Semantic integration** in this context essentially means transforming the information into a graph model of typed nodes (e.g. for products, companies) and typed edges (e.g. for the relationship “company-produces-product”). **Providing unified access** means, letting users in search, explore, visualize and augment the information as if it was from one single integrated system. The user interface can profit from the semantic relationships of the integrated graph to support the user’s search as naturally and intelligently1 as possible. **Semantic technologies** investigated were light-weight graph models (e.g. RDF), ontologies for capturing aspects of the information that can be reasoned with (e.g. RDFS, OWL, F-logic), as well as text analysis technology for detecting content in unstructured text on a higher level of meaning (e.g. named entity recognition).

Mentioning Aletheia is important for two reasons. First, there are quite a number of whitepapers, e.g. from companies or research institutions specialized in ST, which make (sometimes) bold claims about the benefits of ST for enterprises without further substantiating them. This paper targets at evidencing or exemplifying pros and cons of ST based on Aletheia. Second, the author’s experiences in Aletheia, particularly Aletheia’s focus on information integration, have of course shaped his understanding of ST. Thus the discussion in this paper is quite subjective and elides some aspects of ST, e.g. using ST for service description and consumption, which other people might find very relevant. Having said this, this paper does not claim to provide a complete and comprehensive list of the pros and cons of ST: Instead, it should be considered as a subjective compilation of considerations, as “food for thought”, so-to-speak.

The paper is organized as follows. First a (again subjective) definition of ST is provided. After this, one section discusses the benefits and the (not insignificant) costs of ST in enterprise settings are discussed. This is followed by a section dedicated to the ICCS community, before we come to the conclusion.

2 What are Semantic Technologies?

Different communities have different notions of “Semantics” or “Semantic Technologies”. Different communities have a different understanding. For example, people from the NLP (natural language processing) might think of thesauri and taxonomies, database experts might have deductive databases in mind, and software engineers think of UML, the object-oriented paradigm, or model-driven architectures. For this reason, we have to clarify first our understanding of ST.

2.1 Core Semantic Technologies

Under core ST we understand technologies like

- Ontology languages, like RDFS, OWL, or FLogic
- Ontology Editors like Protégé or OntoStudio
- Triple stores and semantic repositories, like OWLIM
- Semantic Middleware, like OntoBroker
- Semantic Frameworks, like Sesame
- Reasoners, like pellet

It should be noted that we do not restrict ourselves to Semantic Web technologies, but include for example FLogic as language and the corresponding applications from Ontoprise like OntoStudio and OntoBroker as well.

2.2 Enablers for Semantic Technologies

The vast majority of information processed by ST is not created from scratch. This applies both to the ST schema (aka ontologies) as well as to the data. Instead, the schema and data in semantic repositories is often based on existing data. Thus a ecosystem of methods and tools is needed which turns existing data or existing documents into semantical information. Such methods and tools can be considered as key enablers for ST.[1]

There are first of all approaches which are so-to-speak directly connected to core ST. Prominent examples are approaches which map relational databases to RDF. A W3c Incubator Group[2] has published in 2009 an overview over such tools in [4]; a very recent report has been published by the research project LOD2 (see [6]). The incubator group has meanwhile turned into a working group which aims to "standardize a language for mapping relational data and relational database schemas into RDF and OWL, tentatively called the RDB2RDF Mapping Language, R2RML", which evidences the importance of these approaches.

There are moreover approaches which have not developed with a dedicated support of core ST into mind, but which are of outstanding importance for ST, namely text mining, information extraction (IE) and NLP approaches. Already in 2006, Timo Kouwenhoven named the following applications:[3]

- information and meaning extraction,
- autorecognition of topics and concepts, and
- categorization.

It is in the nature of IE and NLP approaches that they do not work with 100% accuracy. The success of ST in the long run will (partly) depend on the maturity and accuracy of these tools.

2.3 Semantic Web vs. Semantic Enterprise

It has to be stressed that ST for enterprises is not the same as the Semantic Web, or Semantic Web technologies simply put in place in enterprises. To name (and sometimes overstress) some differences:

- Data: The data in the web is mainly unstructured data (like txt, soc, pdfs) and semistructured data (like html pages or xml files), whereas in enterprises,

[1] This point of view is disputable: The herein mentioned technologies are sometimes understood as genuine ST, e.g. in [5]

[2] http://www.w3.org/2005/Incubator/rdb2rdf/

[3] See http://www.timokouwenhoven.nl/2006_02_01_archive.html.

besides unstructured data, structured data from databases (e.g. ERP systems) is of high importance.

- Domain: The web domain is topic-wise unrestricted, whereas the domain for a given enterprise is restricted to the enterprise business. In the enterprise, sometimes existing business vocabulary can (and should) be reused for ST applications. In the web, we have no unique name assumption and the open world assumption, whereas in enterprises, entities and documents should have only one identifier (thus the unique name assumption can be assumed), and the closed world assumption holds.[4]
- User: In contrast to the web, users in enterprises have specific well-known roles and work in specific well-known contexts. Depending on role and context, the user-access to data and information is controlled.
- Governance: Content-wise, the web is not governed, whereas in enterprises, authorities can govern the vocabularies, content, or the development of ST applications as such.

3 Benefits of Semantic Technologies

ST is said to have various benefits in the context of enterprises. Of course, different authors and different people name different lists of benefits, but some existing or envisioned benefits are frequently reoccurring in different sources. First off all, *data integration* is identified as a key benefit, e.g. by [1, 2, 7, 8]. *Agile schema development* is similarly often mentioned [1, 2, 7, 8]. The first two benefits mainly concern the technical backend of enterprise information systems. For users, *semantic search capabilities* is an often named benefit of STs [2, 7, 8]. Finally, though not directly a feature of ST, *collaborative / social computing* is often brought up as key feature or enabler of ST {2, 7].
In the following, we will elaborate on these four benefits in more detail.

3.1 Data Integration and Federation

„Its the integration, stupid!“ We find this nice quotation in [9], a work which starts with an analysis of the overall enterprise software market ($222.6 billion in 2009 according to Gartner), expresses that ERP (Enterprise Resource Planning) “is still what pays the bills” (ERP has a $67 billion share of the enterprise software market), and examines that “enterprises are all about integration”. They are not alone with this estimation, for [1, 2, 7] data integration is a key asset of ST as well. So we have to dive deeper into the problem, investigate why existing solutions fall short w.r.t. integration, and discuss what ST has to offer.

[4] As with all points in this list, contrasting the WEB OWA and enterprise CWA is disputable. An example for a different point of view can be found in Bergman’s “seven pillars of the open semantic enterprise/”, where he argues for the “open world mindset”. See http://www.mkbergman.com/859/seven-pillars-of-the-open-semantic-enterprise/

The need for data integration and federation is certainly not new to enterprises, even if we look only at enterprise internal, structured data. A common problem for enterprises is the independent development of solutions for the different constituencies, which lead to data being spread across different databases. On the one hand, data is often stored in different formats in different databases, on the other hand, different departments often have a different understanding of the meaning, e.g. semantics, of the stored data. Of course, there exists approaches to cope with this problem, e.g. Master Data Management (MDM) systems which attempt to tame the diversity of data formats (for a short discussion on the shortcomings of MDM systems compared to ST see for example [10]), or Data Warehouses (DW) persist data federated from different databases based on a unified view on the federated data (which is gathered with the well-known ETL-process, including data cleansing and fusion techniques).

Anyhow, looking only at integrating data from different enterprise-internal databases is certainly not sufficient. Besides relational data, other structured data formats have to be taken into account as well, like xml data, Excel files, CAx files, etc. More importantly, more and more valuable information assets are stored in unstructured formats like (the text of) office documents, emails, or (enterprise-internal) forums and blogs. In fact, the ratio of unstructured data amongst all data is estimated to be 80% to 85%, leaving structured data far behind in second place. MDM systems and DWs cannot deal with these kinds of information. With ST, in combination with enabling technologies like text analysis, information extraction and natural language processing, it is possible to integrate such information sources as well.

The need for data federation does not stop at the borders of enterprises: public and governmental data sources become increasingly important. Initiatives like Linked Open Data [11] foster the availability of public datasets with an impressive growth rate in the last 5 years. Even governments encourage (e.g. UK[5] and US[6]) the use and re-use of *their* data-sets as Linked Data[7].

The central goal of open data protocols like Linked Data, OData[8] (Microsoft) and SAP Data Protocol[9] (SAP) is to avoid data silos and make data accessible over the web. Common to all of them is the use of URIs to name things and to provide metadata along with the data itself. While OData and the SAP Data Protocol, which builds on OData by adding a business relevant view on it, favor the relational data model and apply a schema first approach, Linked Data is better suited for the graph data models and supports the schema later approach. The tool support for OData is superior today, while on the other side Linked Data supports semantic reasoning with its web-query language SPARQL.

To summarize the discussion so far: With ST, it is possible to federate data from all relevant sources, independent of its format (databases, XML, excel, CAx, text, etc) and location (internal or external). Federating data is more than just gathering data

[5] http://data.gov.uk/

[6] http://www.data.gov/

[7] http://www.w3.org/wiki/LinkedData, http://www.w3.org/DesignIssues/LinkedData.html

[8] http://www.odata.org/

[9] http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/00b3d41b-3aae-2d10-0d95-84510071fbb8?QuickLink=index&overridelayout=true

from different sources and persisting[10] it in one central repository. Instead, the mutual relationships and connections between the different data snippets have to be embraced. With the graph-based information models of ST, it is not only feasible to provide an appropriate data model for federated information in which those relationships are made explicit. Using the reasoning facilities of ST, it is moreover possible to derive new information from the federated data which is not explicitly stored in any of the sources and which might even be obtained by combining facts from *different* data sources.

In Aetheia, indeed information from different sources like databases, xml-files, excel-sheets and text-documents is federated into a graph-based model, and it is even evidenced in Aletheia that that information is presented to the user which can only be obtained by both reasoning on facts from different data sources.

3.2 Agile Schema Development

Schemata for data and information are usually not very stable: They may evolve over time, the notion of entities and relationships change or are extended, new types of entities or relationships have to be added, whereas other types or relationships might become obsolete and thus are dropped from the enterprise information model.

Enterprises have to cope with the following problems: First off all, a too high fraction of the enterprise data models and business logic is still hard coded in the applications. For this reason, it is error prone and costly to employ changes in the model or business logic. For databases, the situation might look different, as data models and even business rules can be captured by the data model of the databases. But the rise of databases dates back to those times where the waterfall model was prominent in software development, and this is still reflected by the design and execution of relational databases. That is, when a database system is set up, first the conceptual schema of the database is to be developed, which is then translated into the relational model. Only after the relational model has been set up, it is possible to fill the database with data. Even if data models are not hard coded, changes in the data model are costly.

With ST, the situation is different. First of all, due to the high expressiveness of semantic models (i.e., ontologies), it is possible to capture a relatively high amount of the enterprise information model and business logics in the semantical model, which leads to a clearer separation of the application and knowledge model. Secondly, it is not necessary to develop a schema first: It is instead possible to store data in a semantical persistency layer (e.g., a triple store) and add later the corresponding schema information. Moreover, changes in the data model are, compared to relational databases, much easier and can be conducted at runtime. Assuming a smart user interface, this could even be done by end users; the data model of applications can even be extended at runtime by end user with new entities and relationships without breaking the application or requiring it to be re-developed.

[10] To avoid too detailed technical discussions, the question which data has to be materialized in a central repository and which data can be retrieved on the fly from the original data sources is deliberately ignored.

The separation of application and business logic lowers the TCO (total cost of ownership) of applications to a large extent, suits better the state-of-the-art agile paradigm of software development and is assessed to be a key benefit by many professionals (see [1, 2, 7, 8]).

In Aletheia, filling the repository with federated data and the development of the ontologies has been carried out in parallel. Indeed, the ontologies have sometimes been adapted after (informal) evaluations of Aletheia's behavior in the frontend, which evidences the benefits of the agile schema development. Moreover, the separation of application and business logic can be shown with Aletheia as well: For the two main uses cases, namely the use cases from the partners ABB and BMW, the same Aletheia application is used. In this application, it is possible at run-time to switch between the underlying models for ABB and BMW.

3.3 Semantic Search Capabilities

When it comes to the interaction between users and applications, ST have some core benefits as well. First of all, for accessing information, coherent semantic models can be developed which are especially designed for human understanding (e.g. domain- or business-ontologies), and concepts in these models are mapped to the underlying data sources in a manner transparent to the end users. Thus the heterogeneity and complex technical models and the gap between IT and business is hidden. These models can particularly cover the business terminology of the end users, including synonyms (i.e. different terms which denote the same concept) and homonyms (i.e. terms which denote different concepts). (Syntactically) dealing with synonyms is not very complicated (here it is essentially sufficient to maintain lists of synonyms and taking them in use queries into account), dealing with homonyms is more challenging. Homonyms must be resolved. This can either be done by smart algorithms which do that automatically (i.e., in a search for "bank credit mortgage", an algorithm could guess that "bank" refers to financial institutions and not to seating-accommodations) or manually when the user enters search terms. Aletheia deals with homonyms by an autocomplete functionality in its search box (see Fig 1).

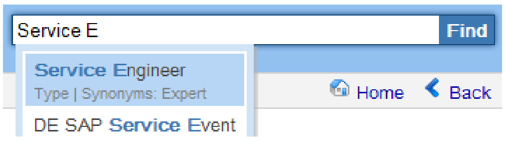
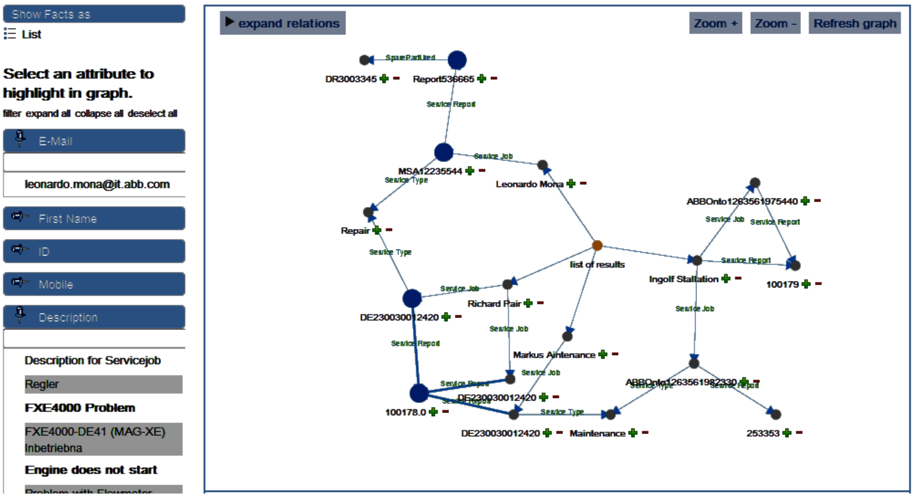

Service E

Fig 1: Autocomplete in Aletheia

"Semantic search" might stand for searching the information space in an explorative manner, or searching *for* specific pieces of information. When it comes to exploring the information space, some user interaction paradigms are quite natural for ST. First of all, the well-known faceted search approach [12] is self-evident. Facets can directly be generated from concept hierarchy of the underlying semantic model. Moreover, as semantic models usually capture relationships between different types of entities as well, a semantically enabled faceted search can allow for navigating along these relationships. Secondly, as semantic data models are usually graphs, graph-based visualizations are similarly natural to employ.[11] The essential idea of such visualizations is to display some entities of the information space as nodes of a graph, and displaying the relationships between these entities as (unlabeled or labeled) edges between the corresponding nodes. In such visualizations, different means to interactively explore the information space can be implemented. For example, the graph can be extended, nodes can be filtered out, subgraphs of interest can be highlighted, etc. Quite interesting is on the other hand to explore for given entities the path between them in the repository.



Select an attribute to highlight in graph.

leonardo.mona@it.abb.com

Description for Servicejob

Regler

FXE4000 Problem

FXE4000-DE41 (MAG-XE) Inbetriebna

Engine does not start

Problem with Flowmeter

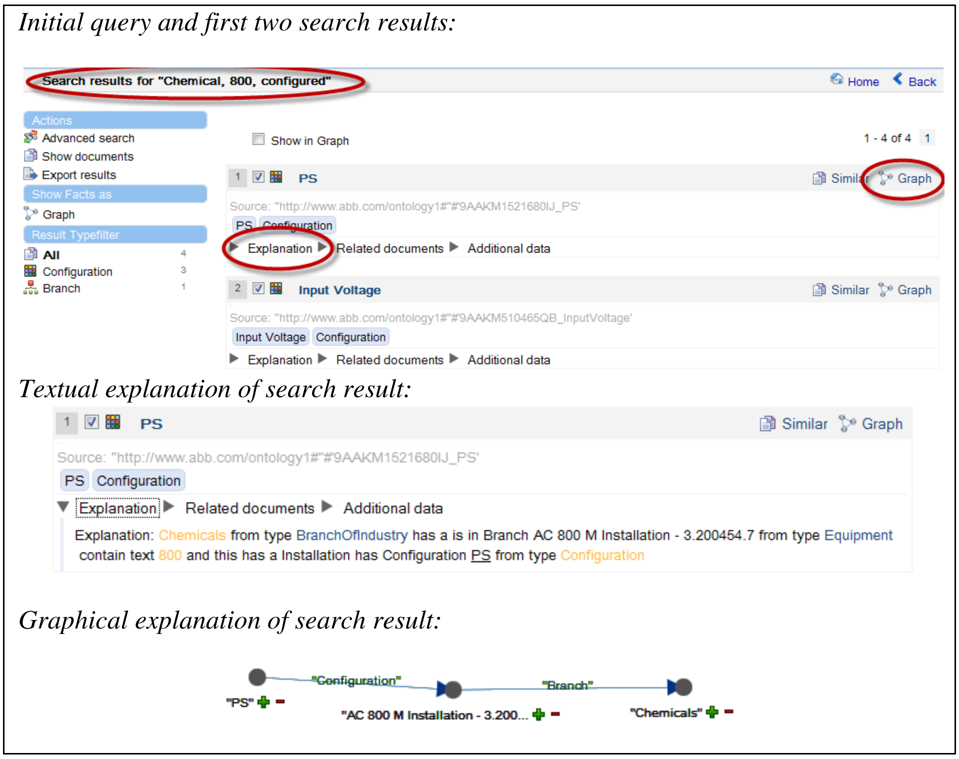
Fig 2: Aletheia graph-based visualization

Besides (more or less) new user interaction paradigms, it is well accepted that a “classical” search box is a key feature requested by users. We have already argued semantic search can embrace synonyms and homonyms, which renders semantic search superior to standard keyword searches. If moreover the structure of the semantic model (e.g. concept hierarchies, relations between entities in the ontology) is taken into account, smart search algorithms can try to understand what the user means with entered text search queries and can therefore provide better search results. Moreover, search results can be personalized, e.g. based on past search or the context of the user.

An example for a smart search is the semantic search functionality of Aletheia. In Fig 3, a user has entered "Chemical, 800, configured" (and during his input, the search terms “Chemical” and “configured” have been disambigued). This search does not search for entities in the search space which are labeled with all three search strings, but for possibly different entities which are labeled with some of the search strings and which are meaningful connected. Indeed, in Fig 3, one search result is provided

[11] On the web are several examples of graph-based visualizations, e.g. google image swirl (http://image-swirl.googlelabs.com/), rel-finder (http://relfinder.dbpedia.org/relfinder.html), or Microsoft academic search (http://academic.research.microsoft.com/).

which is a specific configuration for a product whose description contains “800” (e.g. AC800M) and which belongs to the branch “Chemical”. To understand why search results are found, Aletheia offers both a textual explanation and a graph-based visualization. Both are shown in in Fig 3.



Initial query and first two search results:

Search results for "Chemical, 800, configured"

Source: "http://www.abb.com/ontology1#"#9AAKM1521680IJ_PS'

Source: "http://www.abb.com/ontology1#"#9AAKM510465QB_InputVoltage'

Textual explanation of search result:

Source: "http://www.abb.com/ontology1#"#9AAKM1521680IJ_PS'

Explanation: Chemicals from type BranchOfIndustry has a is in Branch AC 800 M Installation - 3.200454.7 from type Equipment contain text 800 and this has a Installation has Configuration PS from type Configuration

Graphical explanation of search result:

Fig 3: Semantic search in Aletheia

In the long run, the goal is that ST will make the shift from search engines to answer engines by providing what users mean instead of what users say.

3.4 Collaborative / Social Computing

For enterprises, information is a valuable asset, and the benefits of ST discussed so far aim at getting the best out of this asset by integrating all information sources and providing a single point of entry to all information with sophisticated search capabilities. There is anyhow another asset of enterprises which has to be taken into account as well: The enterprises employees and their knowledge. Networking and knowledge exchange between people are becoming increasingly important, thus ST should not only support an integrated access to the data, but attempt to provide an integration of people as well. In the last decade a variety of social network tools has been developed, both in the web sphere (for example, social networking platforms) as well as enterprise internal collaboration platforms (Enterprise 2.0 tools and platforms like semantic media wikis used within companies). So it comes to no surprise that decision makers asses the support of collaboration very important: According to [2], “interoperability is the top priority for semantics; searching/linking information and collaboration rank next in importance – all are top priority for more than half the companies”, and the report states that “as semantically enabled applications come into the Enterprise mainstream, they will bring the integration and interoperability required for next-generation systems, as well as the usability and collaborative features of social computing (“Web 2.0”).”. In fact, in November 2009 Gartner estimated the “content, communications and collaboration (CCC) market” revenue to be at $2.6 billion in 2009[12], and in [15] Gartner states that “a major advance in the Semantic Web, the one that has pushed it along on the Hype Cycle, has been the explosion of social networking and social tagging with sites such as Facebook, YouTube, MySpace, Flickr, Wikipedia and Twitter.”

It has anyhow to be observed that to date, semantically supported collaboration tools (like semantic media wikis) are rare and the CCC market is hardly embracing ST. From an enterprise-internal view, enterprise 2.0 tools cover corporate blogging, intra-enterprise social networking tools, corporate wikis, etc. The danger is that companies might implement various mutually independent enterprise 2.0 services, which would cause information about some objects of interest is scattered over the network of the enterprise. How can ST help here? If the content of enterprise 2.0 tools, like people, objects of interests, content, comments, tags are described by agreed-upon semantics, then enterprise 2.0 tools can better interoperate. This is currently a matter of research.[13] In the web sphere, one expects the convergence of the Web 2.0 on the one hand and semantic web on the other hand to the next web generation called web 3.0. A similar convergence is needed in the enterprise realm for applications which provide a unified view on all enterprise information on the one hand and for applications which support the collaboration amongst employees.

To summarize: in contrast to the ST benefits discussed so far, a semantic approach to collaborative/social computing is still in its infancy and has to to be considered a *prospective* benefit of ST. Promising research on semantically supported collaboration tools is currently conducted, and collaboration tools are likely to be a key enabler for ST (see section 1.3 in [2]).

4 Costs of Semantic Technologies

Semantic technologies do not come for free, there are challenges and cost factors to consider. In the following, the costs and challenges of ST are discussed. This section is based on the findings gained from the research project Aletheia, and a book chapter [13] from Oberle et al which summarizes challenges in adopting ST for software engineering.

[13] names six challenges in adopting ST, namely

[12] http://www.gartner.com/it/page.jsp?id=1223818

[13] See for example the SIOC-project, http://sioc-project.org/.

- "Technical Integration", which includes scalability and performance issues of ST applications as well as the maturity level of ST tools
- "Technical Integration × *n*", which discusses problems if different ST tools are chosen for different use cases
- "Modeling Depends on Use Case", which scrutinizes the cost of modeling ontologies
- "Cost-Benefit Ratio", which shortly investigates the TCO (total cost of ownership) for employing ST
- "How to Measure the Benefits?", which discusses problems to measure the benefits of ST, and
- "Education", which takes the costs for training developers and users into ST into account.

Though essentially addressing cost of ST for software engineering, the findings of [13] apply to other domains as well. The diagram of Fig 4 is taken from [13] and provides an estimation in how specific the above mentioned challenges to software engineering are.

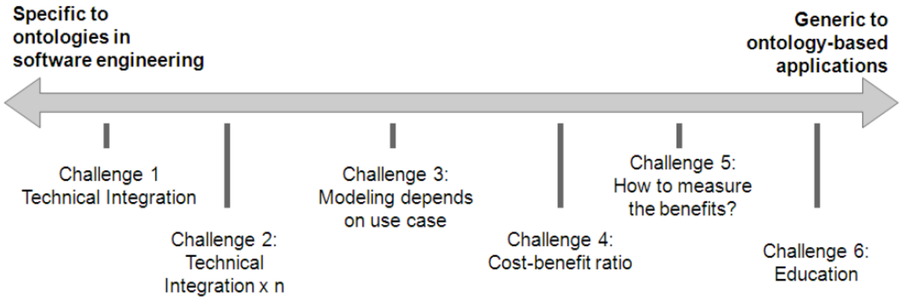

Specific to ontologies in software engineering
Generic to ontology-based applications
Challenge 1 Technical Integration
Challenge 2: Technical Integration x n
Challenge 3: Modeling depends on use case
Challenge 4: Cost-benefit ratio
Challenge 5: How to measure the benefits?
Challenge 6: Education

Fig 4: challenges of ST for software engineering, according to [13]

Essential findings from Aletheia concerning the costs of ST are:

- Performance and scalability issues of semantic middleware,
- complexity of additional technology stack, including training costs for users and maintenance cost, and
- manual operation effort of the ontologies used in Aletheia.

This findings are closely related to the technical integration and modeling challenges from [13].

In the following, we dive deeper into the challenges.

4.1 Technical Challenges

There is a variety of tools for ST available, but most of them are outcome of academic research (e.g. scientific work by PhD students, academic groups or research projects) and thus lacking documentation or ongoing maintenance and further development. For this reason, most of these tools are not mature enough to be used within an enterprise setting and cannot be taken into account when ST for enterprises are discussed.

Of course, this does not apply to all tools and frameworks from Academia (e.g. Protégé has achieved a enterprise-suitable maturity), and there are moreover applications from professional vendors, being it dedicated ST vendors like Ontotext, Ontoprise or Franz Inc. or being it large-scale like Oracle or IBM. But even dealing with mature tools has drawbacks.

Two prominent problems are scalability and performance: ST tools still are a magnitude behind relational databases. Moreover, due to the complexity of the semantic languages (like RDFS, OWL, F-Logic) and the corresponding reasoning facilities[14], it might even happen that the tools never completely catch up. These problems are both mentioned in [13] and experienced in Aletheia.

Moreover, the integration of ST into the IT landscape of an enterprise is challenging. To quote [13]: "Technical integration means the required technology needs to be embedded in the existing landscape of the adopting enterprise. Adaptors have to be written to legacy code and databases, versions of programming languages might have to switched, specific software components might have to be replaced because of license incompatibilities, etc. The challenge typically increases with the size of the legacy code and size of the enterprise's portfolio."

The situation might become worse is different use cases in an enterprise are taken into account. As discussed in [13], different use cases, which often target different beneficiaries, might the use of different ontology languages, editors, stores, and reasoners. Obviously, using different ST tools for different use cases increases the complexity of integrating these tools into the IT landscape. As stated in [13], this probably yields to "the challenge of technical integration might have to be faced × *n*".

Besides these performance/scalability issues and the challenges when it comes to integration, [13] discusses furthermore the challenges

- whether enterprises which want to embrace ST should build ST tools on their own or buy them from third-party vendors, and
- how ontologies are updated in the enterprise IT landscape, which usually offers "a transport system for dealing with updates".

[14] The discussion on the (depending on the languages, sometimes huge) different computational complexities of different languages is outside the scope of this paper and hence deliberately neglected in this discussion.

4.2 Modelling Challenges

Implementing ST in an enterprise use case requires the modeling of an ST schema, i.e., an ontology. Automatic creation of high-quality ontologies is still out of realm, thus ontologies usually have to be manually designed, which is costly and thus increases the TCO (total cost of ownership) when ST are set into place. Anyway, a use case partner in Aletheia understood this effort as an investment beyond Aletheia for their company, as "the ontology can be reused across different software systems and helps the company to maintain a consistent view on their data assets" [AletheiaWP].

Apart from the costs, modeling ontologies is technically challenging as well. First of all, in contrast to the ubiquitous relational model in relational databases, there is a variety of different ontology languages (like RDFS, several OWL profiles tailored for different purposes, or -not being a semantic web language- FLogic) to choose from. Secondly, there is still a lack of mature and comprehensive CASE-tools for modeling ontologies. Finally, there is no standard methodology for ontology modeling. A number of methodologies have been proposed, like "Ontology 101" from McGuiness, the method from Uschold and King, the method from Grüninger and Fox, On-To-Knowledge, the Cyc method, SENSUS, KACTUS, TOVE, METHONTOLOGY, etc. The sheer number of methodologies reveals that this is still work in progress, and no methodology has become accepted as standard methodology.

4.3 Measuring Challenges

The pros and cons of ST are (like in this paper) discussed in a qualitative manner. It is anyhow desirable to *quantitatively* measure the benefits and costs. There are different high-level dimensions which can serve as a basis for evaluating ST:

- Technical dimensions like performance and scalability
- User-centric dimensions like the effectiveness of semantic models for users (e.g., compared to relational models)
- Cost-centric dimensions like the TCO for employing ST

For measuring technical dimensions, particularly evaluation RDF stores, a number of benchmarks have been developed. Well known are the Lehigh University Benchmark (LUBM), which been extended to the University Ontology Benchmark (UOBM) for targeting OWL Lite and OWL DL, and the Berlin SPARQL Benchmark (BSBM).[15] Measuring ST technologies is anyhow inherently complicated, as a huge variety of factors which impact the performance or scalability have to be taken into account. Basic measurements are of course loading and retrieval times of triples in RDF stores, which are impacted not only by the number of triples, but by the underlying schema as well, which might trigger several costly reasoning steps, including the recursive application of rules (e.g. to compute the transitive closure of a relation). Some benchmarks cover mapping of relational data to RDF as well.

[15] More benchmarks can be found at http://www.w3.org/wiki/RdfStoreBenchmarking

So it comes as no surprise that there is indeed a variety of benchmarks, and no benchmark has become accepted as the standard for ST. This contrasts the situation of relational databases, where 1988 the Transaction Processing Performance Council (TPC), being a non-profit consortium of different IT enterprises, has been founded, which defines benchmarks that have become the de-facto standard for databases.

Benefits for users are less concrete, thus evaluating the benefits of ST for users is harder. Indeed, [13] states out that "the ontology and Semantic Web community has been struggling to evaluate their contributions accordingly. Indeed, one hardly finds scientific methods or measures to prove the benefits", but they point out that "other communities share similar struggles". Of course, there are of course quite a number of user evaluations of applications where ST are used, but the problem is the lack of comparisons of these tools to (corresponding) solutions where ST have not been employed. It can be argued that (carefully crafted) semantic models are closer to human model of the given universe of discourse (from a general design point of view, D. Norman argues in [16] for "proper conceptual models"). Apart from cognitive reasons, this argument is even be witnessed by the success of the leading BI company Business Objects (now a part of SAP): the supremacy of Business Objects is based on their patented invention of their so-called "semantic layer", which essentially provides a meaningful, business-user-oriented vocabulary of some domain, which is transparently mapped to SQL queries on relational databases. But still, this argument is of qualitative nature: To the best of the author's knowledge, there are no quantitative evaluations which substantiate the claim that ST are from a user's perspective superior to relational databases.

After discussing the challenges of evaluating the technical and the use-centric dimensions of ST, it remains to elaborate on the cost-centric dimensions. This dimension is usually measured by the total cost of ownership (TCO). In [13], the following formalization of the TCO is used:

TCO ~ TCO drivers × *#stacks in the IT landscape Integration* × *#of technologies*

TCO drivers might include costs for acquiring ST experts or training users in ST, modeling costs, maintenance, and the like. The number of (existing) stacks in the landscape can be explained with the SAP landscape, which features an ABAB sand a Java stack. Finally, the number of technologies refers back to the problem of "technical integration × *n*": If more than one ST editor, store or reasoned is to be used, the factor will increase. But measuring the TCO is not sufficient. Of course, ST can save money as well, compared to other technologies. From an enterprise point of view, a "business case" which captures the rationales for employing ST in an enterprise is needed. As listed in [13], a decent business case must argue for ST

- under consideration of the cost-benefit ratio
- including a deployment plan of available (human) resources
- defining quantifiable success criteria
- proposing an exit strategy
- concerning the business capabilities and impact
- specifying the required investment
- including a project plan
- in an adaptable way, meaning the proposal can be tailored to size and risk.

4.4 Educational Challenges

When ST-based applications are implemented, experts in ST are needed. Being an ST expert requires knowledge in a broad spectrum of topics, e.g.

- knowledge about different ontology languages and their respective capabilities and shortcomings. This particularly includes knowledge about the logical background of (heavy-weight) ontology languages in order to understand the reasoning techniques and capabilities, which often hard for people who lack training in mathematical logic.
- Knowledge about ontology engineering methods and methods, including knowledge about existing ontologies, approaches for re-using ontologies, and methodologies for ontology engineering
- knowledge about existing tools like editors, stores, and reasoners

Thus it takes arguably some effort to become an ST expert.

If an enterprise lacks such experts, either existing employees have to be trained in ST, or ST experts have to be acquired. Even if existing employees are willing to become familiar with ST, teaching them will create considerable training costs. The conclusion from observation can be found in [13]: "Usually, the training costs are very high and managers are not willing to expend them unless there is a compelling business case.”

Of course, acquiring new experts instead of training existing employees raises costs as well. But compared to the ubiquitous relational databases, ST are still a quite new technology and neither established in academia nor industry. Thus it will not only be costly to employ ST experts, it will be harder to find them compared to experts in established technologies like relational databases. Again the conclusion can be found in [13]: “If there is no convincing business case, an enterprise might decide to realize the use case with conventional technologies, i.e., technologies where there is expertise readily available in the company.”

Expenses are not limited to application developers: they are likely to occur for users as well. As discussed in the last section, ST have benefits both in the back- and in the frontend of applications. For the frontend, semantic search facilities have been named. It should be anyhow noted that only in the ideal case, such new capabilities in the frontend are that user-friendly and easy to understand that no educational costs for users have to be taken into account. Such educational costs do not necessarily refer to training courses; self-education (e.g. E-learning) raises costs as well.

5 Buy-in for the Conceptual Structures Community

It is the author’s belief that the conceptual structures (CS) community exhibits significant knowledge for embracing ST, such as theoretical and philosophical background of ST, as well as practical knowledge about

- FCA (both theoretical foundations and practical applications)
- different graph-based knowledge representation and reasoning (like existential graphs, RDF, conceptual graphs in different forms –simple, with

rules, based on different kind of logics, with different levels of negation and context, etc-)

- ontologies (languages, background, modeling)

This knowledge is evidenced both by a significant foundational contributions in terms of scientific papers and books ([17, 18] as well as several, sometimes quite powerful and mature, applications for FCA and CG (e.g. ToscanaJ[16] for FCA, and Amine[17] and Cogitant[18] for CGs).

It is anyhow the author's opinion that the community has not such a significant impact in the field as it might deserve. Other communities do better in this respect. An example is the Semantic Web community, which started later[19] than the CS community, but to this end, it is obviously way more prominent. Besides the amount of scientific work coming from this community, two other aspects are worthwhile mentioning:

- Standards: The SW community managed to (informally or formally) standardize important aspects of their assets. This is best witnessed by the standards set by W3C, but other de facto standards like the OWL API be mentioned as well.
- Projects: The SW community is involved in a huge number of research projects with tangible outcomes, ranging from pure research projects in one scientific institution to the involvement in huge applied research projects with several academic or industrial partners.

Concerning standards, achieving an ISO standard for common logic (CL)[20] has been an important step into the right direction. The CL standard has gained some visibility[21], but is still four years after being established it does not have a strong lobby and is not very often quoted.[22] To the author's opinion, it is not very likely that the conceptual graphs community will gain significantly higher impact or reputation through further standardization activities. A better way to achieve more impact is through conducting or participation in (applied) research projects. A good example are the activities of Gerd Stumme's Knowledge and Knowledge and Data Engineering Group[23], which covers both internal projects which meanwhile gained high reputation (e.g. bibsonomy, which is since 2008 used within SAP as well, which indicates the usefulness and maturity of bibsonomy), or publicly funded projects with partners, like

[16] http://toscanaj.sourceforge.net/

[17] http://amine-platform.sourceforge.net/

[18] http://cogitant.sourceforge.net/

[19] The first Semantic Web Working Symposium has been held in Stanford in parallel to 9th International Conference on Conceptual Structures.

[20] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39175

[21] For example, Pat Hayes keynote at ISWC 09 refers to common logic. Interestingly, additionally he introduces Peirce's ideas of graph surfaces and negation to RDF. See http://videolectures.net/iswc09_hayes_blogic/)

[22] To some extent, this situation can be compared to topic maps, which gained ISO standardization as well.

[23] http://www.kde.cs.uni-kassel.de

Nepomuk.[24] Another example is the recently started research project CUBIST, lead by SAP Research with partners from the CS community.[25] It is the author's belief that the the CS community has relevant expertise concerning ST for enterprises, and a higher engagement in projects would better unleash these valuable assets.

6 Conclusion

"Beware of the hype!" This is a quote found on quite a number of presentations about Semantic Web technologies. Though SW and ST, as discussed, is not the same, this quotation should be applied to ST for Enterprises as well. A well-known approach to describe the maturity, adoption and social application of a given technology are Gartner's hype cycles. According to [15], "since its unveiling, the Semantic Web has been full of promise, but largely unfulfilled. In the last few years this has changed […]", and Gartner refers to the interest of enterprises in ST which caused that change. Gartner rates the benefits of SW high, and it still sees SW at the peak of interests.

As discussed in this paper, ST can indeed fulfill some of its promises. Anyhow, with the turn from academic research to real-world applications in enterprises, a new set of challenges arises. Some of these challenges, like scalability and performance issues or educational challenges are rather general and somewhat ST-agnostic, but for enterprises, it is of great significance to cope with them.

The maturity of ST tools is still not sufficient for enterprises, but it is emerging. Anyhow, even very mature ST tools are likely to fail in meeting some expectations (particularly if these expectations stem from the bold promises made at the dawn of ST), and moreover, new technologies usually do not only solve existing problems, but raise new problems as well. But this will not stop the "semantic wave", the emergence of ST for consumer and enterprise applications. Instead, in the long run, the author expects ST to become one of many mainstream and ubiquitous technologies, both the benefits and the costs will become widely demonstrated and accepted (this is Gartner's "plateau of productivity", the end of the hype cycle of a technology). It is still time to shape ST on its path to become of the bricks in future enterprise IT environments

Disclaimer: Parts of this work have been carried out in the Aletheia project and in the CUBIST project ("Combining and Uniting Business Intelligence with Semantic Technologies"). Aletheia is sponsored by the German Federal Ministry of Education and Research. CUBIST is funded by the European Commission under the 7th Framework Programme of ICT, topic 4.3: Intelligent Information Management.

[24] A list of projects is provided on their webpage, see http://www.kde.cs.uni-kassel.de/projekte

[25] http://www.cubist-project.eu

References

[1] Horowitz, P. (ed.): PwC Technology Forecast, Spring 2009. Available at http://www.pwc.com/us/en/technology-forecast/spring2009/index.jhtml. 2009. Retrieved Sep. 2010.

[2] Final Demand driven Mapping Report. Public report D3.2 from the research project value-it. Available at http://www.value-it.eu, 02/2010. Retrieved Sep. 2010.

[3] SAP Research Dreseden: Lessons on Semantic Information Integration and Access in the Aletheia Research Project. SAP internal whitepaper, 2011

[4] W3C RDB2RDF Incubator Group: A Survey of Current Approaches for Mapping of Relational Databases to RDF. Report, 2009. Available at http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf

[5] Moulton, Lynda: Semantic Software Technologies: Landscape of High Value Applications for the Enterprise. Gilbane Group Report. Available at http://www.expertsystem.net/documenti/pdf_eng/technology/semanticsoftwaretechnologies_gilbane2010.pdf, 08/2010. Retrieved Oct. 2010.

[6] Deliverable 3.1.1: Report on Knowledge Extraction from Structured Sources. Public deliverable from the research project LOD2 - Creating Knowledge out of Interlinked Data. 2011. Available at http://static.LOD2.eu/Deliverables/deliverable-3.1.1.pdf

[7] Stark, A., Schroll, M., Hafkesbrink, J.: Die Zukunft des Semantic Web. Think!innowise Trend Report, 2009. Available at http://www.innowise.eu/Dokumente/Trendreport.pdf, Retrieved Oct. 2010.

[8] West, Dave: What Semantic Technology Means to Application Development Professionals. Forrester Research, Inc Report, 10/2009.

[9] Lunn, B.: Creative Destruction 7 Act Play. Series on semanticweb.com, Available at http://semanticweb.com/index-to-the-creative-destruction-7-act-play_b624, 2010. Retrieved Oct. 2010.

[10] Axelrod, S.: MDM is Not Enough - Semantic Enterprise is Needed. Information Management Special Report, Available at http://www.information-management.com/, 03/2008. Retrieved Sep. 2010.

[11] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R, Ives, Z:. DBpedia: A nucleus for a web of open data.In Proceedings of the 6th International Semantic Web Conference (ISWC), volume 4825 of Lecture Notes in Computer Science, pages 722-735.Springer, 2008.

[12] Yee, K.P., et al.: Faceted Metadata for Image Search and Browsing. In: Proceedings of the conference on Human factors in computing systems, ACM Press, 2003.

[13] D. Oberle et al. “Challenges in Adopting Semantic Technologies.” In Pan, Zhao (eds.): Ontology-driven Software Engineering, Chapter 11. Springer, to appear.

[14] Münchener Kreis: Zukunft und Zukunftsfähigkeit der Informations- und Kommunikationstechnologien und Medien Internationale Delphi-Studie 2030. 2010.

[15] Valdes, R, Phifer, G., Murphy, J., Knipp, E., Mitchell Smith, D., Cearley, D.W.: Hype Cycle for Web and User Interaction Technologies, 2010. Gartner Report.

[16] Norman, D.: Cognitive engineering. In D. Norman and S. Daper (eds): User Centered System Design, p 31-61. Lawrence Erlbaum Associates, Hills-dale, New Jersey, USA, 1988

[17] Hitzler, P., Scharfe, H. (eds.): Conceptual Structures in Practice. CRC Press, 2009.

[18] Chein, M., Mugnier, M.-L.: Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs. Springer, Lindon, 2009