FFI-rapport 2007/02461

Semantic technologies

Bjørn Jervell Hansen, Tommy Gagnes, Rolf Rasmussen, Marianne Rustad and Geir Sletten

Norwegian Defence Research Establishment (FFI)

22 November 2007

FFI-rapport 2007/02461

1085

ISBN 978-82-464-1275-7

Keywords

Semantikk

Semantisk web

Ontologi (Databehandling)

Informasjonsforvaltning

Informasjonsteknologi

Approved by

Rolf Rasmussen Project manager

Vidar S. Andersen Director

Sammendrag

Semantiske teknologier kan beskrives som teknologier som gjør det mulig å bygge mer adaptiv og effektiv software ved å utnytte meningsinnholdet i den tilgjengelige informasjonen. I FFI-prosjektet Semantini, semantiske tjenester i INI (INformasjonsInfrastrukturen) blir disse teknologiene studert for å evaluere om, og hvordan, de kan brukes i militære applikasjoner. Spesielt interessant er det å evaluere hvordan semantiske teknologier kan brukes til å utvikle fleksible tjenester i informasjonsinfrastrukturen i en implementering av det norske militærkonseptet nettverksbasert forsvar.

En av de sterkeste drivkreftene i semantiske teknologier for øyeblikket er arbeidet relatert til Semantisk Web. Semantisk Web er en visjon presentert i (Berners-Lee, Hendler, & Lassila 2001) som er basert på å utvikle dagens World Wide Web fra å være kun en sammenkobling av dokumenter til også å bli en sammenkobling av data. Spesielt forventes semantiske teknologier å bidra til Semantisk Web på to måter: (1) Gjøre flere av dataene som allerede er koblet til weben globalt tilgjengelige og (2) gjøre det enklere for brukere og tjenester å finne tjenester som er koblet til weben og merket med metadata.

I rapporten identifiseres fem kjerneteknologier blant semantiske teknologier: Kunnskaps-representasjon, ontologier, resonnering og regler, spørringer samt agenter og tjenester.

Det forventes at semantiske teknologier vil kunne bidra til å øke potensialet til tradisjonelle informasjonsteknologier når det gjelder å lage mer adaptiv og fleksibel software. I denne rapporten har vi fokusert på følgende områder: informasjonsintegrasjon, fødererte spørringer, mediering, informasjonsfusjon, semantisk tjenesteorientert arkitektur og semantisk grid, semantisk søking samt informasjonspresentasjon og –navigering.

Det finnes fortsatt flere uløste problemer med semantiske teknologier, og noen av disse presenteres her: Problemer relatert til sammenkobling av ontologier, problemer relatert til resonnering og avgjørbarhet, manglende enighet mellom de forskjellige Semantic Web Service-initiativene samt at verktøy og infrastrukturelementer som behøves for å utnytte semantiske teknologier fortsatt er umodne på enkelte områder.

Rapporten konkluderer med at semantiske teknologier er en lovende samling av informasjonsteknologier som potensielt kan levere viktige kapabiliteter også i det militære domenet. Enkelte semantiske teknologier, som f.eks. RDF og OWL, er relativt modne og det dukker stadig opp flere produkter basert på semantiske teknologier på markedet. Flere semantiske teknologier er imidlertid å regne som eksperimentelle, noe som gjør at mer forskning og eksperimentering er nødvendig for å finne ut om de kan levere som lovet og håpet. Trolig vil flere kjernetjenester i den fremtidige informasjonsinfrastrukturen (INI) kunne gjøre nytte av semantiske teknologier, men for å evaluere om disse teknologiene vil kunne ha noen nytte i det norske Forsvaret er det behov for mer inngående studier i samarbeid med militære eksperter.

English summary

Semantic technologies can be described as technologies that provide tools and methods to build more adaptive and flexible software by exploiting the meaning of the information at hand. In the FFI project Semantini, Semantic Services in INI (INformation Infrastructure), these technologies are studied in order to evaluate if, and how, they can be utilised in military applications. Of special interest is their use in providing flexible services within the forthcoming information infrastructure in an implementation of the Norwegian military concept Network Based Defence.

One of the strongest drivers for semantic technologies at the moment is the work related to the Semantic Web. The Semantic Web is a vision presented in (Berners-Lee, Hendler, & Lassila 2001) that is based on evolving the current World Wide Web from a web of documents to a web of data. Semantic technologies are in particular expected to do two things for the Semantic Web: (1) Make more data currently connected to the web globally available and (2) make services connected to the web and tagged with metadata easier to find for users and other services.

In the report, five core semantic technologies are identified: Knowledge representation, ontologies, reasoning and rules, querying, and agents and services.

Semantic technologies are expected to provide several interesting capabilities enhancing the potential of traditional information technologies regarding making more adaptive and flexible software. In this report, we have focused on the following capabilities: Information integration, federated querying, mediation, information fusion, semantic service-oriented architecture and semantic grid, semantic search, and information presentation and navigation.

There are still several issues regarding semantic technologies, some of which are presented here: The ontology mapping problem is not entirely solved, there are issues regarding reasoning and decidability, there looks to be no immediate convergence between the different Semantic Web Service initiatives, and tools and infrastructures for semantic technologies are still immature in some areas.

The conclusion of the report is that semantic technologies are a promising family of information technologies that potentially can deliver critical capabilities also in the military domain. Some semantic technologies, like for example RDF and OWL, enjoy a relatively high level of maturity, and lately there has been an increase in semantic technology-based products brought to the market. However, as several members of the semantic technology family still are experimental, more research and experimentation is needed to evaluate whether they can deliver as promised and hoped. It is likely that several of the core services in the future information infrastructure (INI) will benefit from semantic technologies, but in order to further evaluate these technologies for use in the Norwegian Defence, there is a need for more thorough studies in cooperation with military domain experts.

Contents

1 Introduction

This report is an introduction to the promising field of research called semantic technologies and is a product from the Norwegian Defence Research Establishment (FFI) project Semantini, Semantic services in INI. INI is an abbreviation for information infrastructure, an important part of the forthcoming Norwegian Network Based Defence. The main goal of this project is to evaluate how semantic technologies can be utilised in military applications in general, and in the information infrastructure in particular.

Network Based Defence (NBD) is the Norwegian Defence's adaptation of the American concept Network Centric Warfare (Alberts, Garstka, & Stein 1999). NBD is based upon gaining information superiority that can be used to increase situational awareness, tempo in execution of command and operations, fighting capabilities, survival and self synchronisation. All this will in the end contribute to higher effectiveness. The infrastructure needed to implement NBD is called the information infrastructure (INI), and within this infrastructure we expect semantic technologies to contribute to flexible services that can be used to perform and facilitate information sharing and management.

The term semantics is derived from the Greek word *semanticos* (significant meaning), and is the study of meaning. Originally it is a discipline of philosophy, but in the context of computer science and technology the word semantics points to the use of explicit and formal domain knowledge to make software more flexible and adaptive and thus more efficient. The technologies that are used to fulfil this goal are semantic technologies.

In the areas of distributed systems and knowledge and information management, semantic technologies are expected to solve important problems more easily than what can be done with current information technologies. Examples of such problems are integration of information from heterogeneous sources and interoperability with unanticipated services. Semantic technologies are also expected to provide certain capabilities that would have been difficult, if not impossible, to support without this new collection of technologies. An example of this is automatic composition of services.

One of the strongest drivers for semantic technologies at the moment is the work related to the Semantic Web. The Semantic Web is a vision presented in (Berners-Lee, Hendler, & Lassila 2001), and involves evolving the current World Wide Web from a web of documents to a web of data. Semantic technologies are in particular expected to do two things for the Semantic Web: (1) Make more data currently connected to the web globally available and (2) make services connected to the web and tagged with metadata easier to find for users and other services.

Metadata is a concept that is often brought to the fore when talking about semantic technologies in general and the Semantic Web in particular. There exist several definitions and descriptions of this heavily used term, ranging from the definition '*metadata is data that describes and defines* *other data*' (Wells 2005) to the stance in (Pollock & Hodgson 2004) that '*all IT-systems are metadata systems*'. Taking the former definition as a starting point, metadata provides an important way to encode the meaning of the data.

Metadata can, however, be a troublesome concept. In part this is due to the abundance of different definitions, making it hard to know exactly what metadata means in a given situation. Another problem with metadata is that depending on the definition of data, all data has the potential to be metadata. Whether the data is to be considered data or metadata can depend on its usage.

It is important to note that as this report handles technologies that have a big momentum at the moment, the descriptions herein has to be taken as being our understanding of the technologies at the time of writing.

The expected background knowledge for readers of this report is a basic knowledge of XML, the World Wide Web, Web Services, and service-oriented architecture.

This document is organised as follows: In section 2 we provide a characterisation of semantic technologies and an introduction of their capabilities. Section 3 provides an introduction of what we call the core technologies of semantic technologies: Knowledge representation, ontologies, reasoning and rules, querying, and agents and services. In section 4, we give a short description of the Semantic Web vision, an important driver for semantic technologies, while section 5 describes a selection of areas where semantic technologies are predicted to contribute to solutions. A discussion of possible military applications of semantic technologies is presented in section 6, while section 7 presents some semantic technology issues. The report is then summarised in section 8, and a conclusion is given in section 9.

> Throughout the document, we provide a set of take-aways meant to highlight what we find are the most important points a reader should notice. They are formatted like this.

2 Semantic Technologies

2.1 Characterisation of Semantic Technologies

There are several definitions of semantic technologies in the literature, the following being a selected few:

(Polikoff & Allemang 2004): *A semantic technology is a software technology that allows the meaning of and associations between information to be known and processed at execution time. For a semantic technology to be truly at work within a system there must be a knowledge model of some part of the world that is used by one or more applications at execution time.*

(McComb 2005): *Semantic technologies include software standards and methodologies that are aimed at providing more explicit meaning for the information that's at our disposal.*

(Sheth & Ramakrishnan 2003): *Semantic technology is the application of techniques that support and exploit semantics of information (as opposed to syntax and structure/schematic issues) to enhance existing information systems.*

(Davis 2006): *Semantic technologies are tools. They are used to create, discover, represent, organize, process, manage, reason with, present, share, and utilize meanings to accomplish business purposes. What makes semantic technologies different from previous information technologies is that they represent meanings separately from data, content and application code so that computers as well as people can share, understand and work with them.*

In addition to these definitions, the following definition of semantic interoperability also gives some hints on semantic technologies (highlighted in the text):

(Pollock & Hodgson 2004): S*emantic interoperability is a dynamic enterprise capability derived from the application of* ***special software technologies (such as reasoners, inference engines, ontologies, and models) that infer, relate, interpret, and classify the implicit meanings of digital content without human involvement*** *– which in turn drive adaptive business processes, enterprise knowledge, business rules, and software application interoperability.*

Based on these definitions, we provide the following characterisation representing our view on semantic technologies:

Semantic technologies are software technologies that exploit the meaning of the information at hand and involve the use of an explicit knowledge model.

Examples of technologies that, given the right context, can be considered semantic technologies are ontologies, rule engines, inference engines, and agents.

2.2 Capabilities of Semantic Technologies

Semantic technologies can be thought of as an extension of conventional information technologies. This means that all semantic technologies are kinds of information technologies, but there are information technologies that are not considered semantic technologies. Examples of the latter are XML, databases, and other information technologies where the semantics are not represented explicitly.

To get an understanding of the properties of semantic technologies, we start by observing that the meaning (semantics) of the data is explicitly represented in addition to the data itself and the program logic. This is the major difference between semantic technologies and traditional information technologies. Further, this representation can be understood by both humans and computers. This makes it possible for computers to share and work with the semantics with minimal, or in some cases no, human intervention.

The loose coupling of semantics, data, content, and application code can be seen as a natural continuation of a trend that has been apparent since the 1980s (see also Figure 2.1): Gradually, more and more application elements have been extracted from the main program. In the 80s, this trend started with the extraction of data from the applications into databases, giving a loose internal data coupling. This trend continued into the 90s when the World Wide Web, with its connections between unstructured documents, provided a loose global document coupling. Next, in the 00s, service-oriented architecture (SOA) and business process modelling (BPM) extracted processes from the programs providing a loose process coupling. The natural next step is what semantic technologies provide; the extraction of the meaning of the data as well as the rules from the programs providing a loose coupling between applications, data, and the meaning of the data. This latter coupling means that applications built using semantic technologies are much more flexible regarding new ways to communicate, new ways to encode data, cooperation with unanticipated applications etc. This also has the effect that developers can spend less time on coding applications, and thus have more time to make even better models of the problem domain.

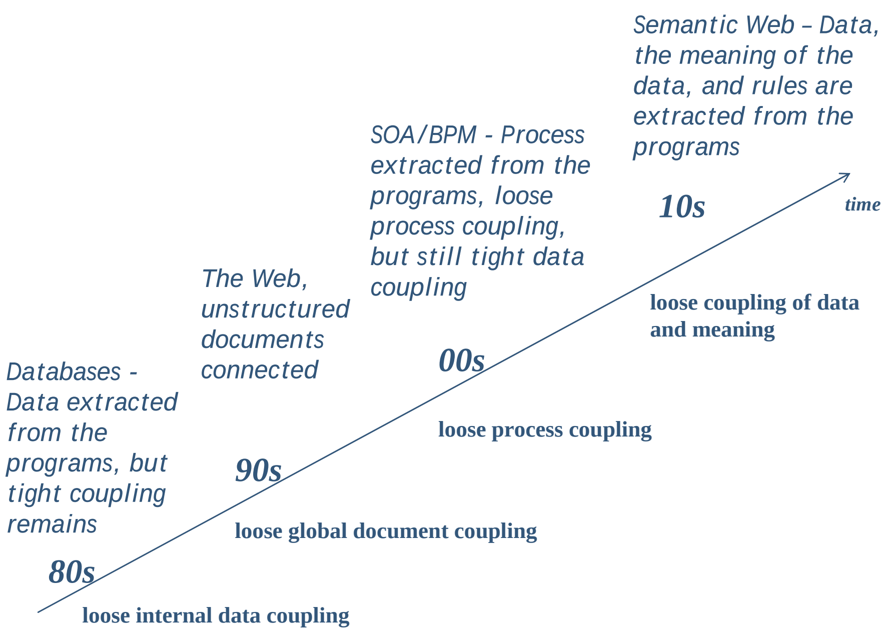


Figure 2.1 The gradual loosening of the coupling between the different elements of applications (Norheim 2007)

At this point it is worth pointing out that most of the problems that semantic technologies can address, can also be solved by conventional information technologies. One example is the handling of semantics. A conventional application can indeed be capable of handling semantics, but the semantics will have to be explicitly programmed by developers. This will, in its turn, make it very costly to change the program to e.g. fit new information exchange needs. In contrast, an application built using semantic technologies will have the semantics represented outside of the application code. This makes the latter category of applications more suited to handle changing requirements. For one thing, a change in the semantics doesn't necessarily force a recompilation of the source code. Reasoning is another example. Once again an application built without using semantic technologies will have to resort to hard-coding to solve the problem, losing the flexibility and adaptability of an application built using semantic technologies. The reason for this difference is that semantic technologies offer a separation of the reasoning apparatus and the program logic, making it possible to change the reasoning apparatus without changing the program logic.

By making the semantic models that systems are built on explicit and formal, mapping or linking such models becomes easier, and may create the effect of a common data model even though it does not actually exist. By defining queries on this model, one can define views on the information available. In other words, information becomes easier to combine and repurpose, and therefore becomes more adaptable.

Semantic technologies also allow the information specified according to various schema to coexist in the same storage space. This makes this family of technologies more open-ended, and allows runtime extension of data models and schema.

Documents like (Davis 2004), (Polikoff & Allemang 2004), and (Davis 2006) present long lists of both proven and (mostly) expected capabilities of semantic technologies. As a first taste of the expected capabilities of semantic technologies, here is a condensed list from these sources:

- Integration and fusion of information from semantically heterogeneous sources,
- federated querying,
- translation,
- inferring new information from an information set,
- dynamic discovery, invocation and orchestration of services,
- Semantic Web Services,
- service negotiations (e.g. quality of service),
- information presentation,
- information filtering (for example based on geographic area),
- semantic search,
- information navigation (Topic maps, portals), and
- semantic wikis.

A selection of these capabilities is described in more detail in section 5.

> Compared to traditional information technologies, semantic technologies offer tools to ease the making of more adaptable and flexible information and software.

3 Core Technologies

In this section we give a description of what we find are the core semantic technologies. Although they are tightly related, we have separated them into knowledge representation, ontologies, reasoning and rules, querying, and agents and services as illustrated in Figure 3.1.

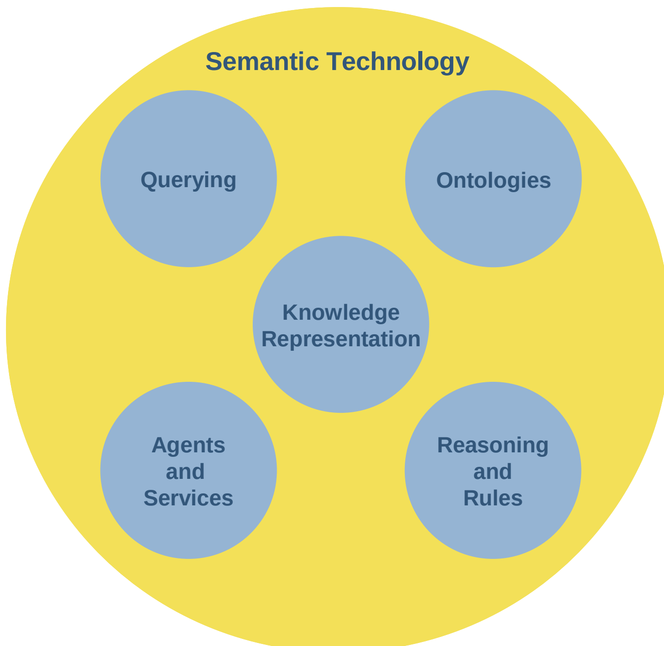


Figure 3.1 The core semantic technologies

The technologies are not necessarily semantic in general. Technologies like rules and querying, for example (see sections 3.3 and 3.4, respectively), are not considered to be semantic unless they are used with a semantic knowledge model.

3.1 Knowledge Representation

Knowledge representation is a field in computer science that has its roots in both the cognitive science research community and the artificial intelligence (AI) community. The aim is to represent and store knowledge and make it accessible to computers, which in turn can infer new conclusions from existing knowledge. This is achieved by basing knowledge representation languages on logic.

Humans tend to relate different pieces of knowledge. The equivalent of this in the knowledge representation field is called a semantic link. This linking needs a representation language that allows continuous addition of new knowledge. Often, a network structure is used to represent this (e.g. a semantic network in cognitive science).

In the stack of Semantic Web standards, the Resource Description Framework (RDF) (W3C 2004b) is a graph-based data model for representing knowledge. It has, amongst others, an XML serialisation, which makes it platform-independent, just like all XML. In RDF, so-called triples are stored. A triple is a subject-predicate-object tuple, where subjects are resources, and predicates specify the relationships between subjects and objects. Putting triples together result in a graph. An XML-document, on the other hand, is a tree. An illustration of the difference between XML's tree-based model and RDF's graph-based model is shown in Figure 3.2.

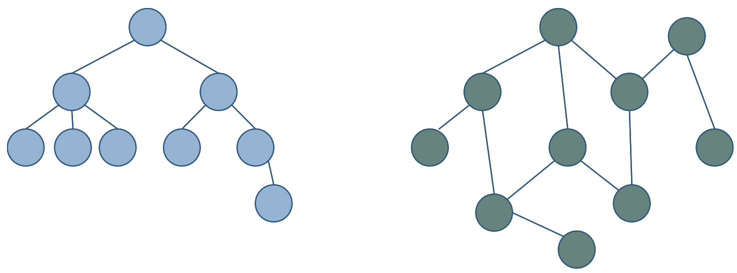


Figure 3.2 XML's tree-based model (left) compared to RDF's graph-based model (right).

Description Logics (DL) (Baader et al. 2003) is a family of logic-based knowledge representation formalisms often encountered in semantic technologies. The area of DL dates back to the early 1980s in an initiative to use fragments of first-order logic to provide semantics to network-based knowledge representation structures like semantic networks and frames. One feature of DL is that it makes a distinction between general knowledge about the problem domain (concepts) and the knowledge specific to a particular problem (instances/individuals). Because of this division, a DL-based knowledge base can be divided into two components: A terminology box (TBox) for the concept definitions constituting the general knowledge, and an assertion box (ABox) for the instances constituting the knowledge specific to the individuals of the domain.

Knowledge representation aims at representing and storing knowledge, making the knowledge accessible to computers. One of the most important models for representing knowledge is the graph-based Resource Description Framework (RDF).

3.2 Ontologies

An ontology is defined in (Studer, Banjamins, & Fensel 1998) as: A formal, explicit specification of a shared conceptualisation. The terms are explained as follows:

- *Formal*: The ontology should be computer readable. This means natural languages are excluded.
- *Explicit*: The type of concepts used and the constraints on their use are explicitly defined.
- *Shared*: Reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.
- *Conceptualisation*: Refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon.

In other words, ontologies are formal semantic models of a domain of interest, and are used to define the concepts used to describe an area of knowledge as well as the relations between these concepts.

A good way to understand what ontologies are is to compare them to other IT models. Here is one such comparison made by Irene Polikoff of TopQuadrant and quoted in (Davis 2006):

- Like databases, ontologies are used by applications at run time (queried and reasoned over). Unlike databases, relationships are first-class constructs.
- Like object models, ontologies describe classes and attributes (properties). Unlike object models, ontologies are set-based and dynamic.
- Like business rules, they encode rules. Unlike business rules, ontologies organise rules using axioms.
- Like XML schemas, they are native to the web (and are in fact serialised in XML). Unlike XML schemas, Ontologies are graphs not trees, and can be used for reasoning.

Ontologies can also be linked to the more commonly known terms vocabularies and taxonomies, as shown in Figure 3.3. Here a taxonomy is described as a vocabulary with structure, while an ontology is characterised as a taxonomy with relations, limitations, and rules.

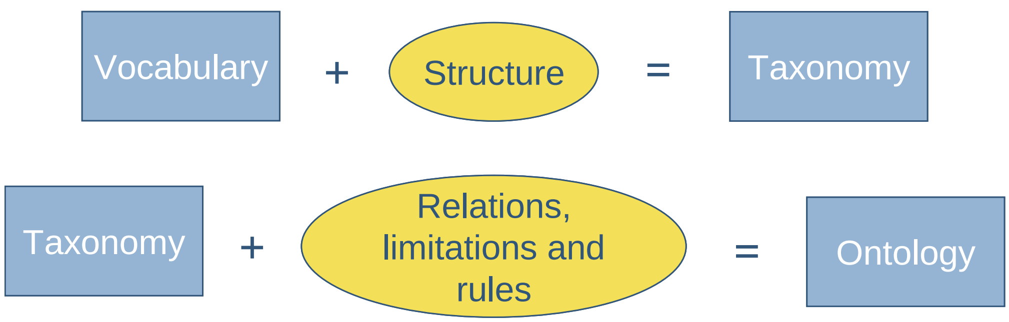


Figure 3.3 The link between vocabularies, taxonomies and ontologies

But even with a definition and suitable characterisations, it is often hard to tell whether a model is an ontology or a traditional data model. One way to evaluate whether a model of interest does indeed classify as an ontology, is to evaluate it according to the expressiveness of the modelling language it is based upon. The more expressive the modelling language, the more the model belongs to the ontology side of an ontology spectrum reaching from simple word lists up to ontologies based on very expressive logics. This is illustrated in Figure 3.4. Usually the boundary between ontologies and models that are not considered ontologies is drawn somewhere between 'Thesaurus' and 'Conceptual model', see for example (McGuinness 2003).

There exist several languages to specify ontologies, but the dominant language today is the Web Ontology Language (OWL) (W3C 2007f) from the World Wide Web Consortium (W3C). OWL is a description logics-based language and comes in three flavours with increasing expressivity (as also indicated in Figure 3.4): OWL lite, OWL DL, and OWL full.

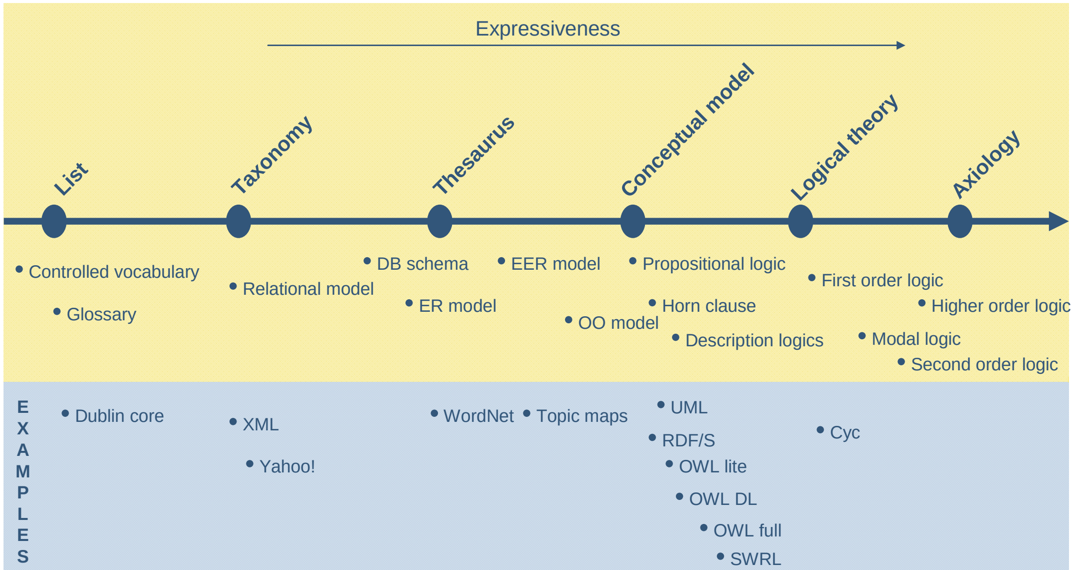


Figure 3.4 Expressiveness shown in an ontology spectrum. Based on (Davis 2006)

OWL allows us to create ontologies that describe and restrict concepts (classes) and properties. The latter can be either object properties that link individuals, or datatype properties that link individuals to data values. Using OWL as an ontology description language, the definitions of the concepts and the description of the individuals can be held within the same file. This is true even if OWL is based on description logics where concepts are handled in the TBox and individuals in the ABox (see section 3.1). But due to the fact that individuals can be described using plain RDF while the concept definitions often require the expressivity of an OWL flavour, one can often find ontologies divided into two files: One file for the concepts expressed in OWL, and one for the individuals expressed in RDF.

One of the key benefits of using an ontology-based approach to develop and integrate data models is that ontologies can be linked to each other. An ontology can import other ontologies and refer to concepts defined there. Hence, it is possible to develop ontologies in a modular and component-based way. This is illustrated in Figure 3.5 with an example taken from the gene ontology project (Stephens et al. 2006). Component-based ontology development can be done both in a top-down and a bottom-up way. The top-down approach is based on extending concepts from an existing, higher-level ontology, whereas a bottom-up approach is based on linking to a higher-level ontology. The bottom-up approach may be used when integrating models that were originally developed in isolation.

Often, when different models have been developed in isolation, it is necessary to bridge between semantically equivalent concepts that are syntactically or structurally different. This is called ontology mapping, and is the key to solving interoperability issues on the semantic level. There is, however, no current best practice on how to do this kind of mapping. This topic is further explored in section 7.4.

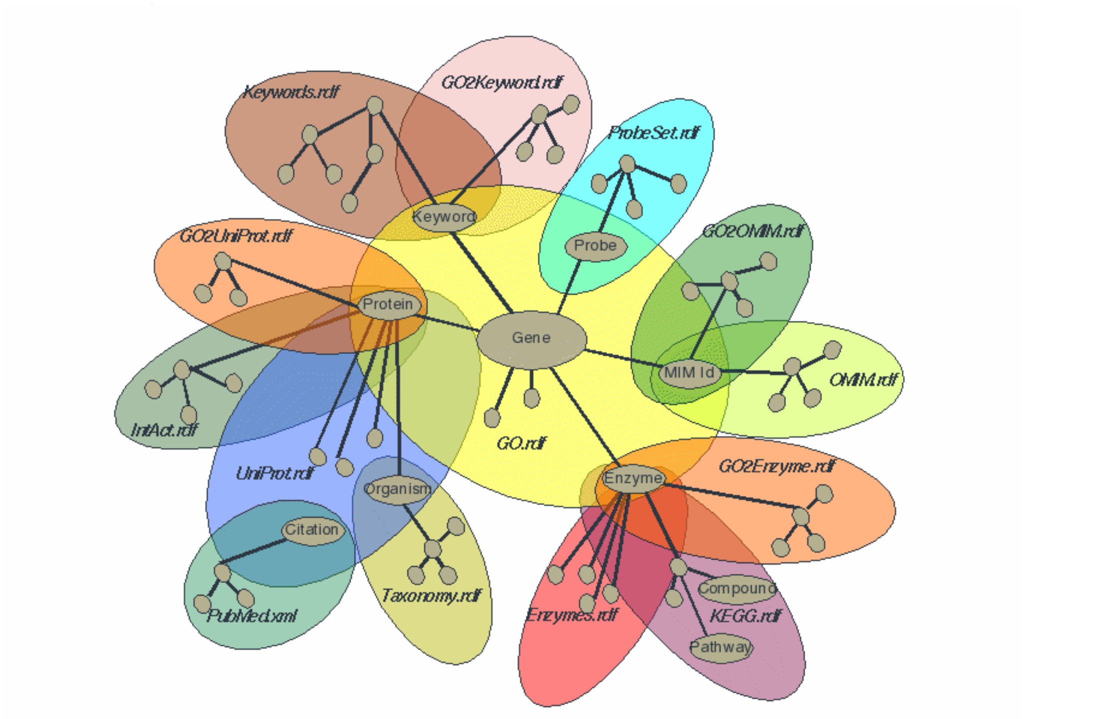


Figure 3.5 Linking ontologies instead of creating one large ontology

Another important aspect of ontologies, is their role as a formal framework for metadata. This can provide the metadata with a high level of precision, as the meaning of the terms will be well defined. Further, ontology linking can also be advantageous in the context of metadata as it provides a formal way to bring metadata from different domains together in much the same way as different ontologies can be brought together.

Ontologies are explicit and formal semantic models of a domain of interest. Ontologies can be linkable, and OWL is the DL-based web ontology language of W3C.

3.3 Reasoning and Rules

The purpose of reasoning in connection with semantic technologies is twofold:

- Ontology design assistance (e.g. consistency checking).
- Inferring new information in an ontology-based knowledge base.

An example of reasoning to assist a user in Ontology design is checking whether the ontology is consistent, i.e. checking that the concept definitions do not lead to any contradictions. As for the inferring new information case, the inference is done using a set of inference rules and the information already present in the knowledge base. The inference rules can be either the axioms of the ontology, custom rules provided by a user, or both. In the latter case, the reasoning is called hybrid reasoning.

The reasoning can be performed based purely on the ontology axioms, i.e. the fundamental rules of the logic the ontology is built upon. In the case where the ontology is based on a description logic, which is the case with OWL and which we will consider here, this is called subsumption reasoning: Determining whether a concept D is considered more general than a concept C, which would make every instance of the concept C also an instance of concept D.

However, using only the ontology axioms as basis for reasoning provides a user with limited expressivity regarding what can be modelled and later be used as a basis for computing inferences. A well known example of this in OWL is the fact that without custom rules there is no way to infer that an individual B is the uncle of individual A based on the fact that B is the sibling of an individual C which in its turn is the parent of A (see Figure 3.6). Added expressivity can, however, be obtained by introducing a rule language to allow the user to provide rules. This also corresponds well to the idea that rules play an important part in encoding knowledge in a domain.

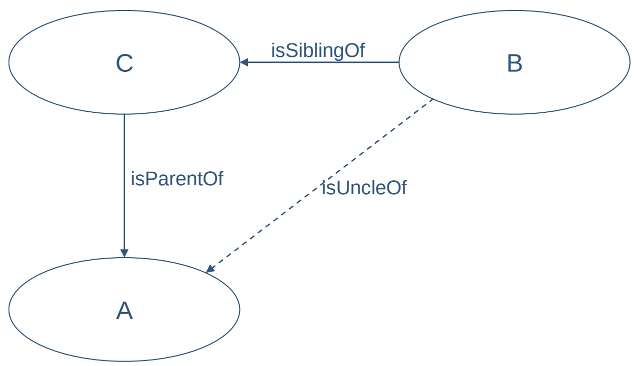


Figure 3.6 The OWL uncle example

An important feature regarding reasoning in semantic technologies is the ability to create generic reasoners that can do automatic reasoning with a wide range of applications. Several such reasoners exist, for example Racer (Haarslev & Möller 2001), FaCT++ (Tsarkov & Horrocks 2006), KAON2 (Motik & Sattler 2006), and Pellet (Sirin et al. 2007).

At the time of writing, the semantic technology community has still not been able to agree on a rule language standard. There are several initiatives to date, the most prominent being the Semantic Web Rule Language (SWRL) (Horrocks, Patel-Schneider, & Bechhofer 2005) and the Rule Markup Language (RuleML) (Boley & Tabet 2001).

Within the W3C's Semantic Web activity there is also an ongoing activity on standards for rules specification in the Rules Interchange Format (RIF) (W3C 2007c) initiative. Here, rather than specifying one rule language to become a W3C recommendation, the strategy is to specify a core format to allow rules written in different languages to be published, shared, and re-used between different applications.

Reasoning in connection with semantic technologies provides the users with ontology design assistance and gives the capability of inferring new information from a knowledge base. In order to increase the restricted expressivity of ontology specification languages, there is a need for user-defined rules.

3.4 Querying

Querying is a fundamental way of collecting information from information systems. In systems based on semantic technologies, information is typically represented using knowledge representation methods (see section 3.1) and is typically being stored in some kind of knowledge base. As one of the most important knowledge representation formalisms is RDF, a query language that works with graphs of triples is needed. The main W3C effort in this area is SPARQL Protocol and RDF Query Language (SPARQL)(W3C 2007e). As the name implies, SPARQL is a collection of recommendations for a query language, a query protocol for accessing RDF data over a network, and a result format for returning result sets. SPARQL is mainly used to access instance data, or ABox data in the DL terminology. The SPARQL query syntax has many similarities with the database query language Structured Query Language (SQL) and has four query forms:

- SELECT – returns a table of results.
- CONSTRUCT **-** returns an RDF graph, based on a template (an RDF equivalent to XSLT – extensible stylesheet language transformations).
- ASK – boolean query.
- DESCRIBE – returns an RDF graph chosen by the query processor.

In addition, optional matches and filters can be used to increase or limit the number of result bindings.

The protocol which specifies the behaviour of a remote query endpoint can be both HTTP and SOAP-based. It has one operation, which is the query operation.

The result of a SPARQL query is wrapped in what is called an XML Query Result Format. One (complete) result is returned for each possible query match. A JavaScript Object Notation (JSON) format has also been proposed.

It is possible to query knowledge bases in a way similar to querying databases. SPARQL is the W3C query language and protocol for querying RDF.

3.5 Agents and Services

A central concept in the initial thoughts about the Semantic Web was the use of agents. Agents are software components able to take advantage of computer-readable elements from semantic descriptions. The US Defence Advanced Research Projects Agency (DARPA) funded work on the Semantic Web trough the DARPA Agent Markup Language (DAML), and as the name implies DAML was a language for agents. OWL in its turn was derived in part from DAML. For agents to fully take advantage of the Semantic Web with its current technology stack, they have to be able to understand OWL ontologies.

Figure 3.7 shows how agents are envisioned to run without constant human involvement to accomplish user-defined goals. The user defines a goal and an agent achieves the goal by gathering, filtering and processing computer-readable information. We define agents as autonomous, proactive and social. They are autonomous as they are able to make decisions on behalf of the user, proactive in the sense that they may take initiative when appropriate, and social as they interact with other agents in order to complete their tasks.

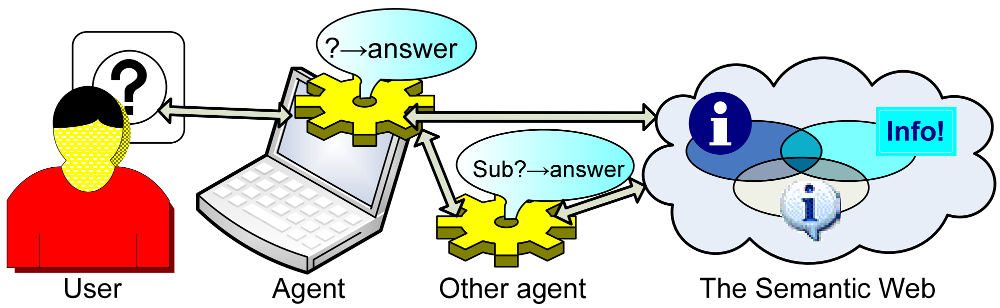


Figure 3.7 An agent on the Semantic Web, delegating tasks to another agent on behalf of a user

In addition to handling information, agents are expected to be of great value in the context of Web Services. By combining Web Services and semantic technologies there is a potential for achieving more dynamism and automation in service environments. Web Service technology provides a solution to the protocol heterogeneity problem of distributed services. Semantic technologies make it possible to describe services in a computer-readable way, allowing data heterogeneity to be solved at the semantic level. Web Services that are enriched with semantic service descriptions are called Semantic Web Services.

All standard concepts needed for describing the properties and capabilities of a Semantic Web Service are defined in a service ontology. Some typical concepts used to describe a service are input, output, precondition and effect. Input is what the service expects to receive from the user, output is what the user can expect to receive from the service, precondition is what is expected to be true before the service can be used, and effect is the state of the world after the service is finished. As an example of a precondition, a service description may state that a user has to have a valid credit card. Similarly, the effect of the service could be that the balance on the user’s bank account is changed.

The individual service descriptions are instances of the service ontology described through concepts from other domain ontologies. E.g. if a service uses position as input to a service, the concept position is defined in an ontology separate from the service ontology. This combination of a domain ontology and a service ontology is used when discovering Semantic Web Services.

Figure 3.8 shows how agents use the service ontology to discover services, compose complex services, and invoke services to accomplish goals specified by the user. For this to be possible, service providers describe services and users define goals/requests according to the concepts of the service ontology. The agent uses the request and the descriptions to reason about services and to match the request to a service description. A complex service might require the agent to invoke several services in order to fulfil the goal. Invocation is done by the agent and resolves the user's goal. This is called service composition.

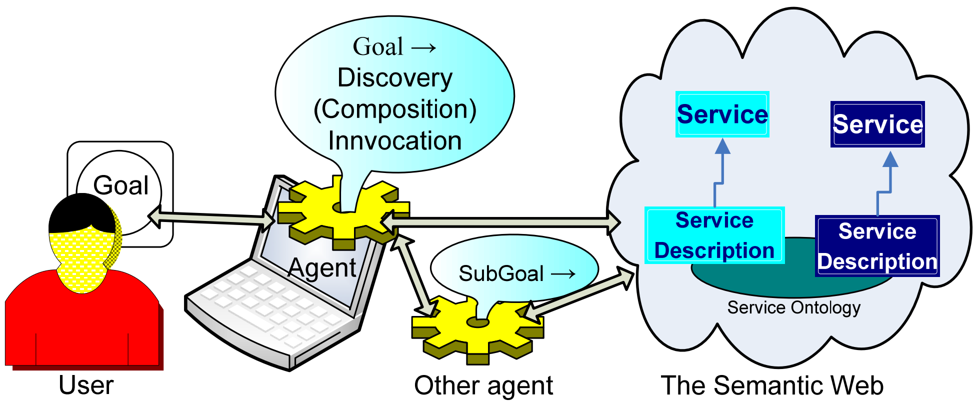


Figure 3.8 Agents utilising Semantic Web Services described using the same service ontology

Figure 3.9 shows how linking between ontologies is used to make syntactically interoperable services semantically interoperable. Here the only way for the consumer to understand the service description (SD) expressed using ontology B, is to do a linking between this ontology and its own ontology A. We discuss this in more depth in section 5.1.

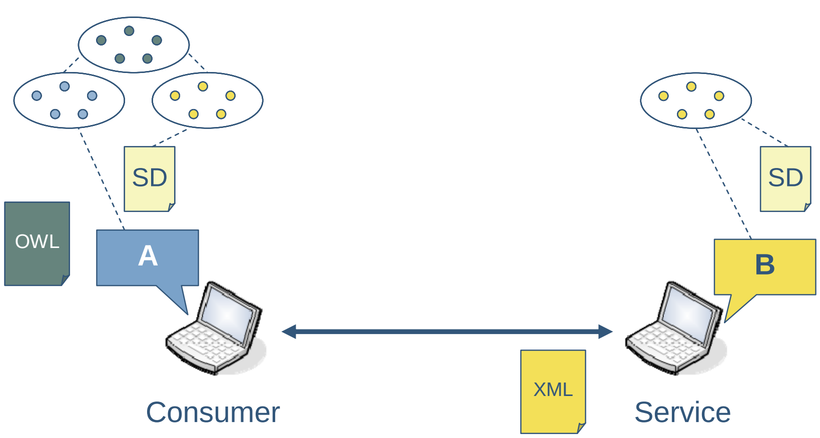


Figure 3.9 Using linking of ontologies to bridge between syntactically interoperable services

Figure 3.10 shows how ontologies are used to perform matchmaking, i.e. matching between a consumer's service request (SR) and a provider's service description (SD) in a service registry. The service provider describes the service according to the domain ontology and the service ontology (not shown). The consumer expresses his request knowing the concepts of the service ontology and having his own domain ontology. Again, by exploiting linking between concepts in different ontologies, the registry can compute that the services can communicate using the mechanism in Figure 3.9. The request and the service description are matched followed by an invocation of the service. Before the service is invoked there might be some form of negotiation between the parties about e.g. quality of service.

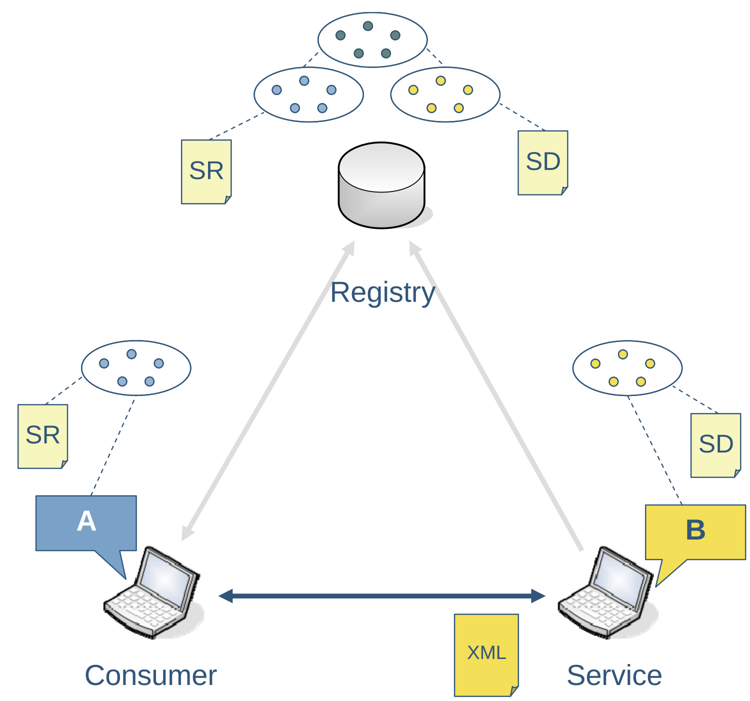


Figure 3.10 Semantic service discovery and matchmaking

There have been, and still are, a number of research efforts in the area of Semantic Web Services. Figure 3.11 shows the timeline of development for the different initiatives. There are differences between them both regarding how they view service semantics and regarding what logical language best represents these semantics. Some initiatives extend existing standards while others offer a mapping to existing standards.

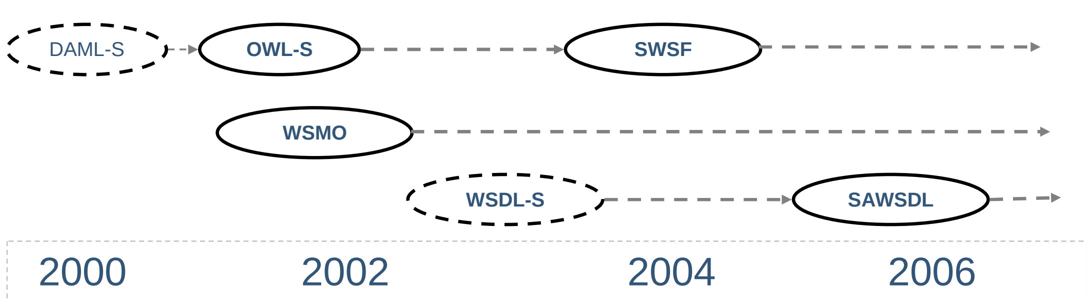


Figure 3.11 The Semantic Web Services initiatives

As may be seen in the figure, the main initiatives currently are:

- Web Ontology Language – Services (OWL-S) (W3C 2004a), based on work founded by the US Department of Defence in the DARPA Agent Markup Language (DAML) project.
- Web Service Modelling Ontology (WSMO) (Roman et al. 2005), founded by the European Commission, Science Foundation Ireland and the Vienna city government.
- Semantic Web Services Framework (SWSF) (W3C 2005a), a part of the DAML program.
- Semantic Annotations for WSDL (SAWSDL) (W3C 2007d), derived from WSDL-S (Web Service Semantics). WSDL-S is a project from IBM and The University of Georgia.

Below, we present these initiatives in more detail.

OWL-S is the most mature initiative and uses OWL constructs in its definition of a service ontology. In addition to OWL constructs, other logical languages are used for representing service behaviour. The drawback is that these additional logical languages are not modelled as part of the ontology but must be embedded in the OWL-based service descriptions and be processed by e.g. rule engines.

WSMO is considered OWL-S' competitor. The work on WSMO was motivated by what was identified as shortcomings in OWL-S and OWL with respect to service modelling where OWL-S uses logic outside the ontology. To remedy the problems, the WSMO initiative designed an ontology language and a framework for service descriptions using logic they considered to better represent services. WSMO promises some level of convergence with OWL-S by adding a logical layer corresponding to the lowest form of logic used in OWL in their specification.

SWSF aims at embracing elements from OWL-S, WSMO, Business Process Execution Language (BPEL), Process Specification Language (PSL), SOA, and service choreography. Semantic Web Services Language (SWSL) (W3C 2005b) is the ontology language used to define the service ontology. Instead of choosing one logical approach, SWSF has chosen to develop two versions of both the ontology language and the service ontology to accommodate different types of logic. The concepts in the ontologies match and are translatable in weakened form.

SAWSDL has recently been accepted by the W3C as a recommendation. As a counterweight to the increasingly complex Semantic Web Service specifications, SAWSDL takes a more pragmatic approach by adding semantic annotations to WSDL files using external ontologies. There is no restriction on the ontology language used to specify the concepts the services exchange. This makes it possible to use both existing and future external ontologies and to describe a service according to multiple ontology languages.

> Agents are software components able to take advantage of computer-readable elements in information and services. Semantic Web Services are Web Services described using ontologies, facilitating automation of several tasks.

4 The Semantic Web

4.1 The Vision

It is fair to say that semantic technologies have become revitalised due to the vision of the Semantic Web, pioneered by the founder of the World Wide Web, Tim Berners-Lee (Berners-Lee, Hendler, & Lassila 2001). In this vision, web content is annotated according to ontologies and the whole web becomes one computer-processable knowledge base. In addition to the annotating of web content, the Semantic Web vision is also a vision of a change in the World Wide Web from a web of documents, as it is today, to a web of data. In other words: With the Semantic Web it will be easier for users (and applications) to get access to data from databases, electronic calendars, etc., and reuse these data. The vision of the Semantic Web has often been criticised as being “Artificial Intelligence all over again”. However, being pragmatic, surely some advantages can come from using this kind of technologies.

In developing the components of the Semantic Web, the World Wide Web Consortium (W3C) provides technologies and standards that can also be used within and between enterprises and large organisations. This is true independently of the success of the Semantic Web.

4.2 The W3C Semantic Web Standards

The stack shown in Figure 4.1 shows the most recent version of the so-called Semantic Web layer cake, which illustrates W3C’s take on the pieces needed in the Semantic Web puzzle.

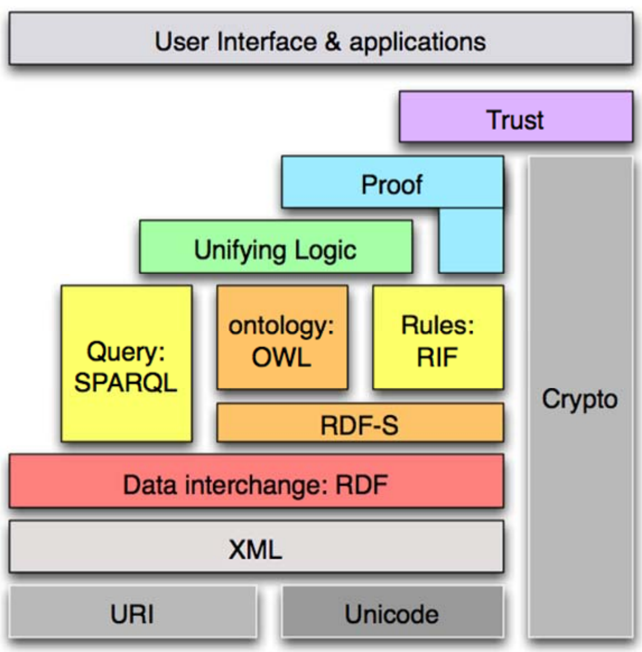


Figure 4.1 The stack of semantic technologies for the Semantic Web - the Semantic Web Layer Cake

Building on Uniform Resource Identifiers (URIs) and XML, RDF (see section 3.1) is the format for representing information in the Semantic Web. It can be queried using the SPARQL query language for RDF (see section 3.4) and restricted using RDF Schema (RDF-S) (W3C 2004c) or the more expressive OWL (see section 3.2). Additionally, there is work in progress on a Rule Interchange Format (RIF), which will be used to exchange rules between different rule systems. The upper levels (as well as the crypto block) do not have any abbreviations associated with them. This is because the technologies in this area are not ready yet.

In the literature, the semantic technologies developed to facilitate the Semantic Web can be seen denoted as Semantic Web technologies. The two terms 'semantic technologies' and 'Semantic Web technologies' are often used interchangeably, but in (Sheth & Ramakrishnan 2003), Semantic Web technologies are defined to be semantic technologies that adhere to the standardisation being done by W3C in order to facilitate the Semantic Web. Following this line of thought, W3C standards like RDF, RDF-S, OWL, and SPARQL are examples of Semantic Web technologies, while a semantic technology like Topic Maps is not considered a Semantic Web technology. In this document, however, we do not make this distinction, and thus use semantic technologies and Semantic Web technologies interchangeably.

The Semantic Web is an extension of the current web where the use of knowledge representation techniques is envisioned to allow computers to collect data on the web on behalf of humans.

4.3 Annotating Resources on the Semantic Web

In order to realise the vision of the Semantic Web, the resources on the World Wide Web have to be annotated, i.e. labelled with metadata. This is true for both documents and data that reside on the web.

The metadata terms can be formally defined in ontologies, providing a high level of precision as the meaning of the terms will be well defined. This can be considered a top-down approach to metadata: A group of domain experts, or another authority, defines which terms are available for use. The advantage of an approach like this is, in addition to the terms being well-defined, that it opens up the ability to do reasoning (see section 3.3) on the metadata.

There is also the bottom-up approach where the users apply their own metadata terms without having to consider a well-defined vocabulary. The result of such a metadata creation strategy is often called a folksonomy. The use of folksonomies is exemplified in popular online services like del.icio.us (del.icio.us 2007) and flickr (Flickr 2007).

One important technique for annotating web pages is microformats. Microformats are a set of re-usable standards that help people and applications share commonly needed information over the Web. Examples of common microformats are hCalendar (microformats.org 2007a) for handling events in calendars and hCard (microformats.org 2007b) for handling contact information.

The two leading W3C initiatives regarding annotating the Semantic Web are RDFa (W3C 2007b) and Gleaning Resource Descriptions from Dialects of Languages (GRDDL - pronounced "griddle") (W3C 2007a). The latter initiative has recently reached the status of a W3C recommendation.

RDFa is a syntax that expresses the metadata using a set of elements and attributes, embedding RDF in HTML documents in a way that is imperceptible to the user of a standard web browser. This is done by adding the RDF as metadata in the tags of the HTML document, making the document metadata computer-readable. The RDFa proposal is managed by W3C, and has a working draft status at the time of writing.

The idea of GRDDL is to take advantage of available microformats to create RDF-based annotations for web pages automatically. This is done by a GRDDL transformation, which has to be specified either in the document itself or in the microformat profile.

5 Selected Capabilities

This section presents a selection of the capabilities listed in section 2.2. The selection reflects the project's view at the time of writing of the most interesting capabilities of semantic technologies, and is illustrated in the outer circle in Figure 5.1.

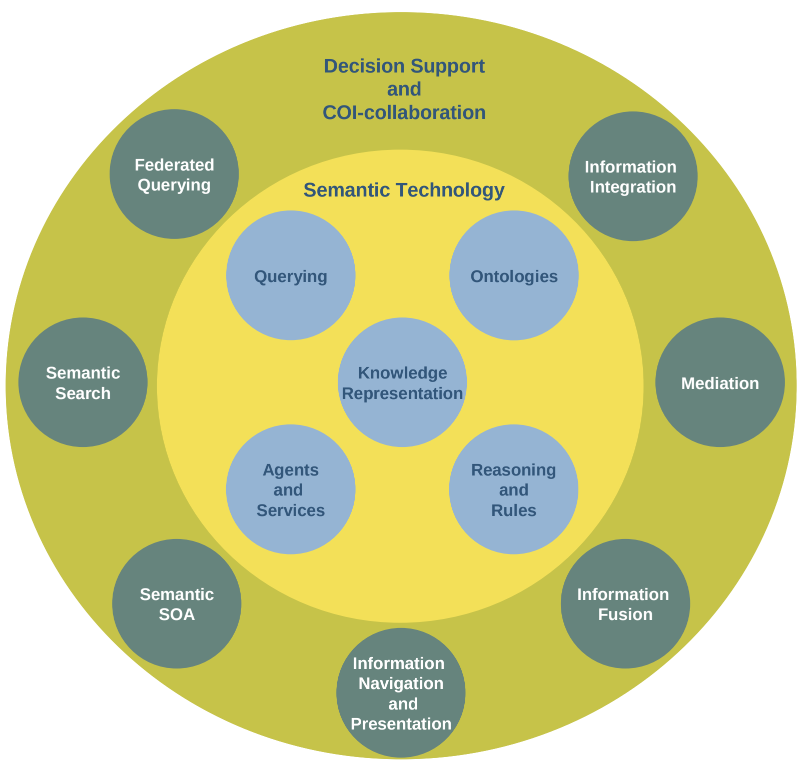
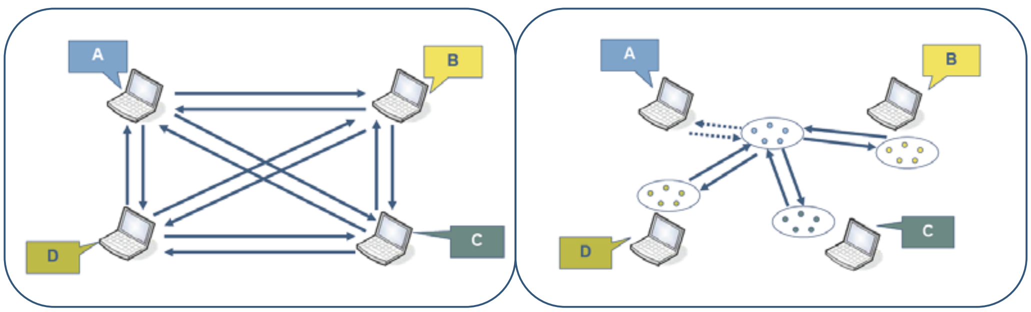


Figure 5.1 The selected capabilities shown together with the core technologies

5.1 Information Integration

The main idea of semantic information integration is to solve semantic mismatch between different formats and systems at the semantic level. By separating the ontologies from the systems, one can define mappings between these ontologies. Since ontologies are formal models based on logic, it is possible to use automatic reasoning tools on these models to execute mappings. This can be used to integrate information from heterogeneous sources automatically (based on the specified mappings). The result is a model-driven way to resolve semantic mismatch, which introduces a (logical) hub-spoke structure at the semantic level.

By doing information integration in this centralised or federated way, proponents of semantic technologies claim that it is possible to approach linear growth in the number of integrations that must be carried out. This contrasts vastly with the current worst-case scenario of exponential growth in such integrations, since it is based on a decentralised (point-to-point) integration structure (shown in Figure 5.2 (a)). The exponential growth in integrations is commonly known as the n^2-problem, which is roughly the maximum number of integrations that are needed for n systems. With the semantic information integration model, it is claimed that only n integrations are needed at best (shown in Figure 5.2(b)).



(a) (b)

Figure 5.2 Point-to-point transformation (a) vs. centralised/federated (b) information integration

The hypothetical benefits of semantic information integration therefore clearly are interesting, since most large organisations suffer from having many different systems that should ideally share more information between them. Additionally, new and existing collaboration partners could often be provided with more or better information. The semantic approach is claimed to make information more adaptable, better supporting evolving and changing requirements for information sharing.

Another benefit of semantic information integration is the ability to integrate information from heterogeneous sources as different as web pages, word processing documents, databases and Web Services. Using RDF as the standard way to represent data, the data can be integrated and repurposed using querying and possibly mapping and reasoning mechanisms. See Figure 5.3 for an illustration of this.

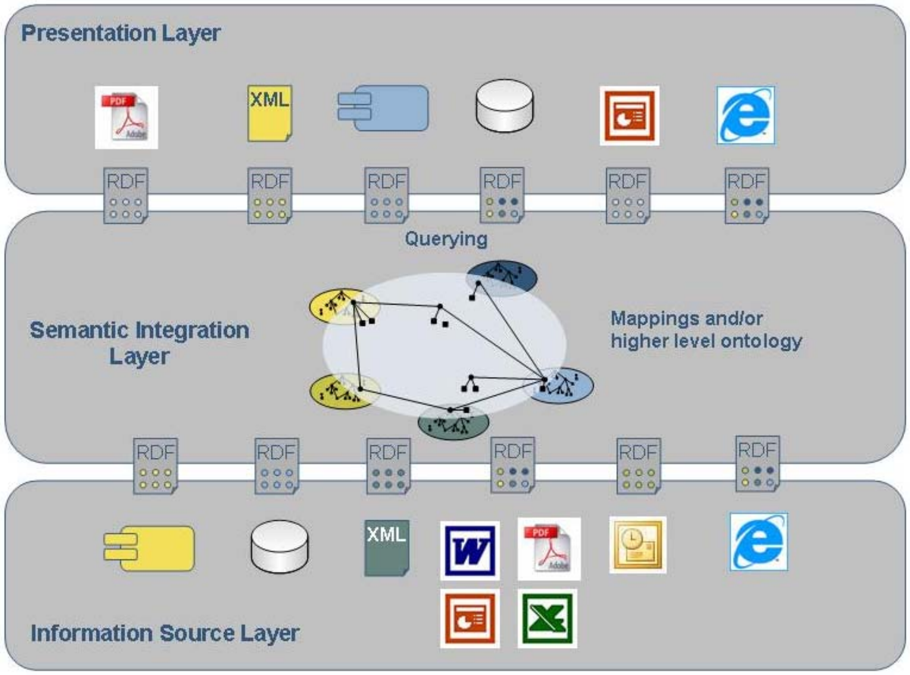


Figure 5.3 Semantic information integration can bring together data from heterogeneous sources.

The main idea is to

- create ontologies based on the source formats,
- create a higher-level bridging ontology,
- map the source ontologies to the bridging ontology, and
- design a query that operates on the bridging ontology.

By using different reasoning mechanisms to execute the mappings specified, automatic translation between the native models and the bridging ontology can be executed in runtime. As a result, the query specified will provide answers from all the sources.

Semantic information integration provides a more scalable way to integrate information from heterogeneous sources, based on automatic reasoning and mapping between ontologies.

5.2 Federated Querying

A topic related to information integration (see section 5.1) is federated querying, also called federated search or federated data access. While not a new concept, this pattern is definitely facilitated by semantic technologies. The problem that federated querying seeks to solve is that important or relevant information may reside in different databases in different organisations or agencies. Thus, these databases have different ownership and information needs to be preserved and updated in its respective organisation. Still, sometimes the only way to get the needed information or answers is by combining information from these different databases.

The idea of federated querying is to pose a query to a central ‘engine’, which in turn may translate this query and pose sub-queries to the different databases. By combining the results of such distributed query execution, new information previously not available may be obtained. This clearly is beneficial in a system-of-systems environment. There may be variations on this theme, by e.g. caching information at the federated query engine, feeding results from the first query into the second query and so on. The value that semantic technologies add to this pattern is the use of the triple model for storing data, as well as ontologies. This facilitates the integration (possibly using reasoning) of the different query results into one result. Figure 5.4 shows an example of a federated query used to deliver information to a web page. The federated query result could of course also be provided as a service to an automated process on the client side.

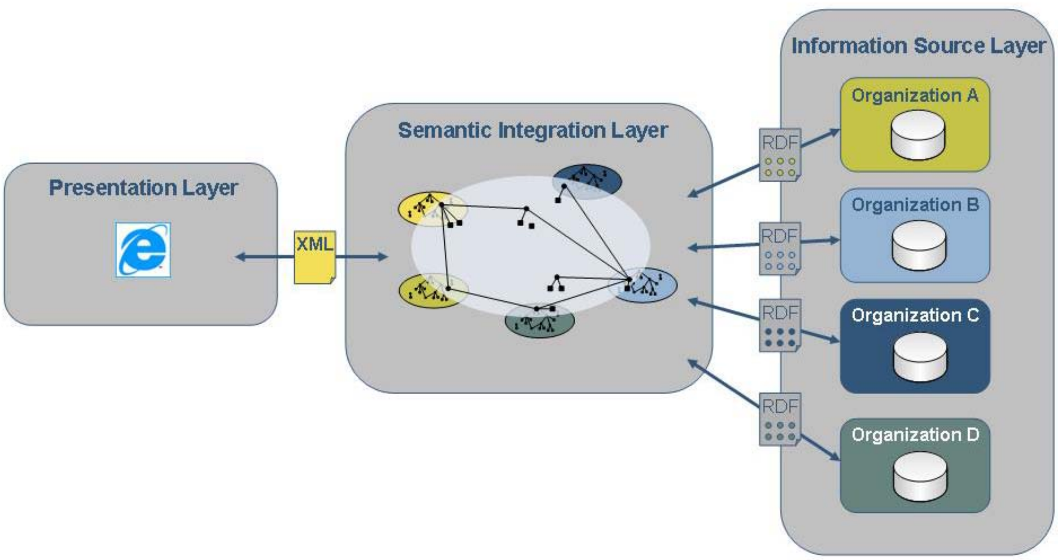


Figure 5.4 A federated query spanning four different organisations.

The federated query pattern may be viewed as information integration in pull, rather than push, mode and is facilitated by using RDF. The main reason for this is that RDF data can refer to various ontologies, meaning that triples referring to completely different models can coexist in and be added to the same knowledge base. Since SPARQL also specifies how to query a remotep endpoint, it is well suited for this kind of tasks. There exist experimental extensions to the current SPARQL protocol and implementations that seek to facilitate federated querying.

Semantically enabled federated querying utilises semantic information integration to query databases in potentially different organisations while preserving database autonomy.

5.3 Mediation

Mediation is usually defined as the process of reaching an agreement between two parties. In the context of this document, we can talk about mediation as resolving mismatch between services or agents in terms of formats, protocols and languages. Often the mediation process will be taken care of by one or more components called mediators. These mediators may be transparent to the parties between which the mediation is taking place.

Format mediation is resolving format mismatch between two components, and is closely related to information integration (see section 5.1) in the sense that it usually will involve information integration in addition to formatting the output to an appropriate format. At the same time, format mediation inherits the benefits of information integration. An example of format mediation is shown in Figure 5.5 where a client with a format A and a service with a format B want to exchange information. This is not possible without a mediator in-between to translate between the two formats. Semantic technologies' contribution in this is basing the mediation on mapping between ontologies. That means allowing the translation to be specified declaratively on the semantic level rather than syntactically as one-to-one mappings between the components.

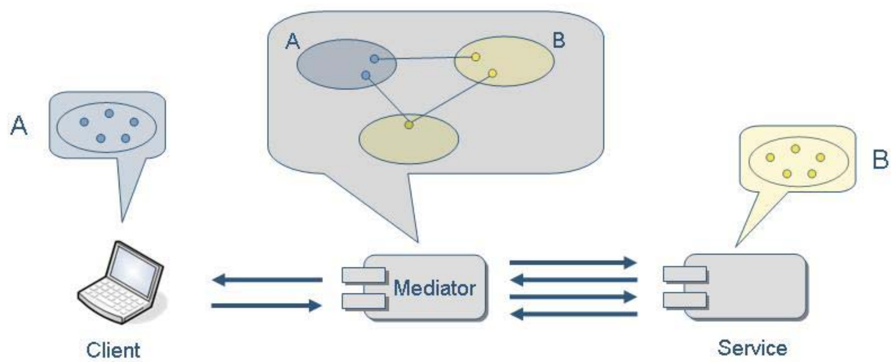


Figure 5.5 Format and protocol mediation

With protocol mediation the challenge is related to interaction between a client and a service, especially between a client and a service that were not predicted to interact. As an example, in Figure 5.5 the client expects two interactions with a service, while the service in question expects four interactions. In this case, this makes straightforward cooperation between the client and the service impossible. The role of the mediator, then, is to make sure that the client and service can perform the interactions they are designed to do. The mediator can be based on hard-coding, but semantic technologies are envisioned to contribute to a more generic planning of what the mediator needs to do to facilitate interaction between the client and the service. An example of this can be using automatic reasoning based on a planning domain ontology to assist the planning.

The distinction between format and protocol mediation is not always clear-cut, and we envision that often, both will have to be used together to solve certain mediation problems.

Language mediation, or translation, using semantic technologies is based on exploiting a semantic level representation of text in different languages to do translation. This can be based on the idea of labelling concepts with language tags, as can be seen in Figure 5.6. There the concept ‘Cat’ has labels with language tags: One tag for the Norwegian representation of the concept (“Katt”), and one for the French representation (”Chat”). If a system then somehow is able to relate the string “Katt” in a text to the concept Cat, it is possible to follow the relation ‘label:fr’ to the French label “Chat”. Thus we can have a translation from Norwegian to French. As can also be seen from the figure, this is related to the research area natural language processing (NLP). There will obviously be uncertainty attached to the process of tying a string to a concept, but such NLP considerations will not be pursued here. Further information on NLP can be found in e.g. (Jurafsky & Martin 2000).

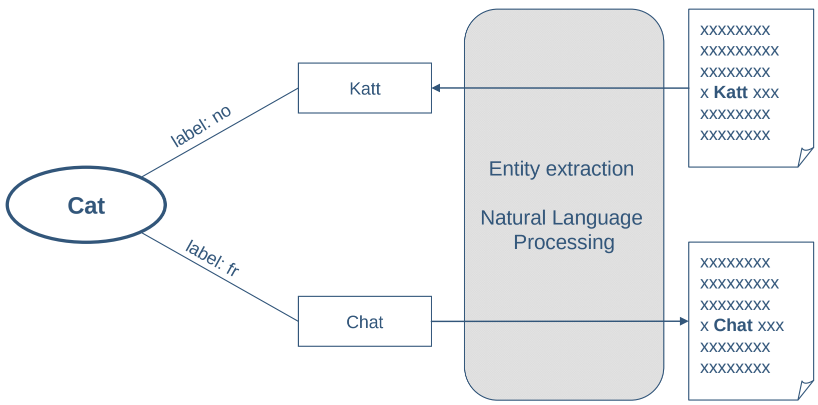


Figure 5.6 Translating the Norwegian string ‘Katt’ to the French string ‘Chat’ via the concept ‘Cat’

Semantic meditation uses semantic technologies to resolve semantic and/or protocol mismatch between clients and services, possibly in a transparent way.

5.4 Information Fusion

While the two terms information integration and information fusion are often used interchangeably in the literature, in this document we make the following distinction between the terms: Information integration is the merging and aligning of information from different sources, while information fusion in addition to information integration also includes the process of inferring new information based on the integrated information set. This difference is illustrated in Figure 5.7.

The term information fusion is in its turn often used interchangeably with data fusion, which can be defined as: *a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of greater quality will depend upon the application* (Wald 1999).

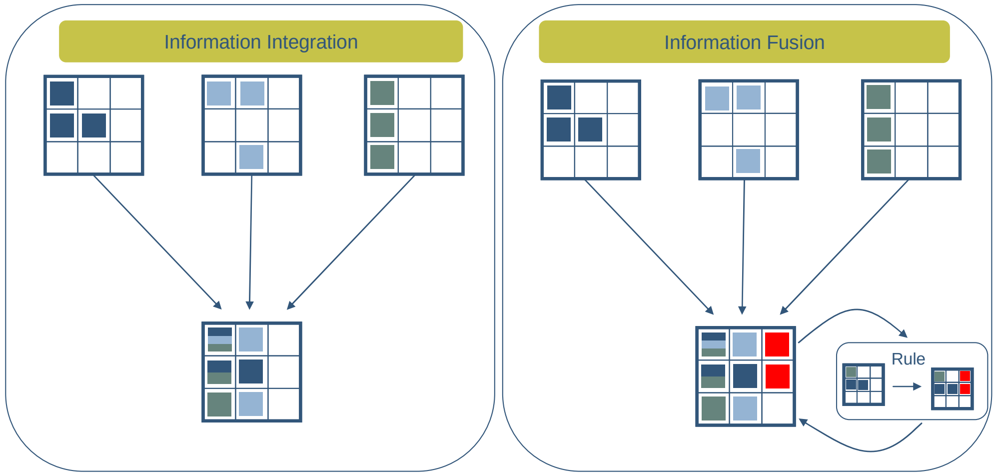


Figure 5.7 The difference between information integration and information fusion. Whereas information integration only merges and aligns the information, information fusion in addition infers new information

There exist several models for data fusion, but the most commonly used, at least in the military domain, is the so-called Joint Directors of Laboratories (JDL) data fusion model (Hall & McMullen 2004). In the latest edition of this model, the data fusion domain is divided into six levels, as illustrated in Figure 5.8. Often, levels 2, 3, and partly 4 of the JDL model are called high-level data fusion or information fusion. In this document we adopt the latter naming convention.

Whereas data fusion on levels 0 and 1 concerns itself with estimating single objects and their states, information fusion concerns itself with estimating relations between the single objects. Symbolic reasoning is needed in order to estimate the relations, and it is at this point that semantic technologies enter the stage. As relations are directly expressible in semantic technologies like RDF and OWL, there is reason to believe that these kinds of technologies can offer solutions when dealing with information fusion problems.

Another important aspect of estimating relations is that they are not directly observable. In order to estimate them, one solution is to use user-defined rules as a basis to do reasoning (see section 3.3). This can be seen as a way to emulate the manual process of looking for patterns in the information to make hypotheses about entity relationships. If this information is described according to one or more ontologies, semantic technologies offer generic reasoners that should be able to infer these relations.

Moreover, in situation analysis, which in an information fusion context includes level 2 data fusion (situation refinement), the goal is to provide the user with situation awareness: *The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future* (Endsley & Garland 2000). This formulation relates well to the goal of semantic technologies to handle the meaning of information. The highlighted parts of the following definitions on level 2 and 3 data fusion from (Steinberg & Bowman 2001) further strengthens the tie between information fusion and semantic technologies: *Situation Assessment* (level 2) *is the estimation and prediction of entity states on the basis of* ***inferred relations*** *among entities*, whereas *Impact Assessment* (level 3) *is usually implemented as a prediction,* ***drawing particular kinds of inferences*** *from Level 2 associations.*

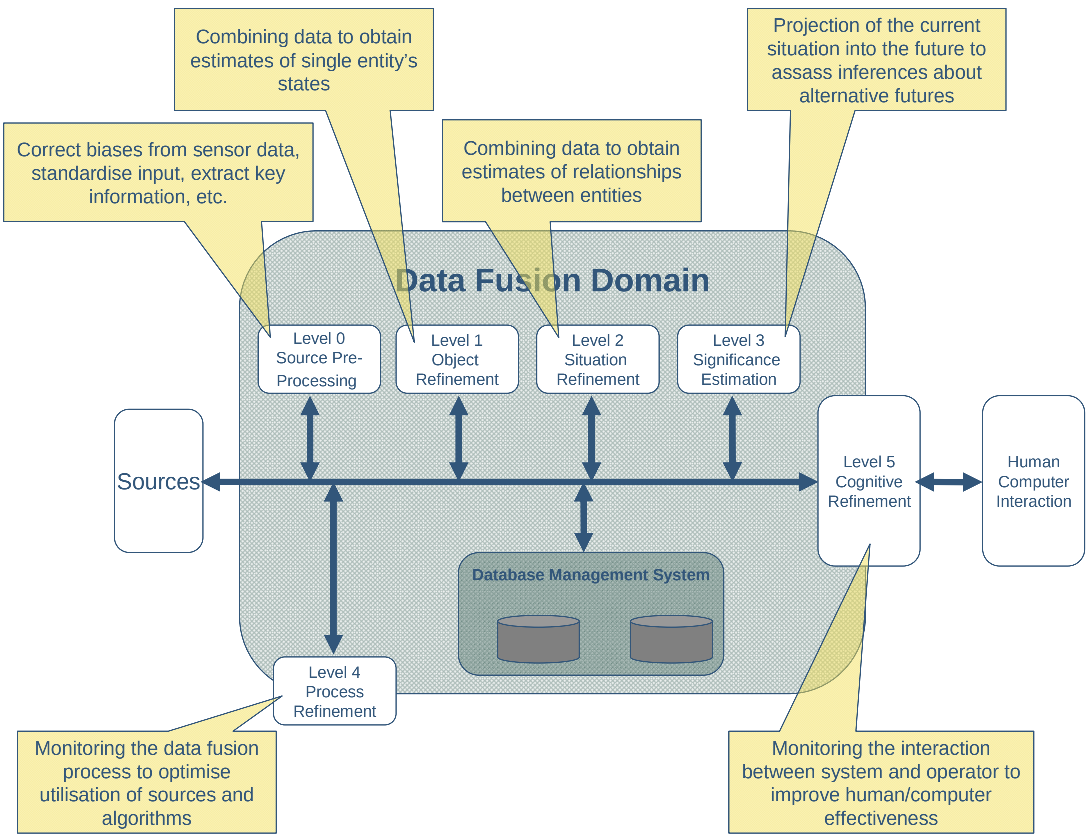


Figure 5.8 The JDL data fusion model with its six levels

A possible scenario of semantic technologies in information fusion is illustrated in Figure 5.9:

1. A user defines rules to estimate relations. Relations of interest in a military context could be for example `canThreaten`, `canFireAt`, etc.
2. Information from heterogeneous sources is collected and stored in a knowledge base in RDF form.
3. A generic reasoner is fed with the user-defined rules and reasons on the information in the knowledge base. It infers new information related to the relations of interest – the new information is shown as light grey RDF triples in the figure and will show up as a result of querying the knowledge base.
4. The feedback to the user could for example be warnings about certain (undesired) relations being detected.

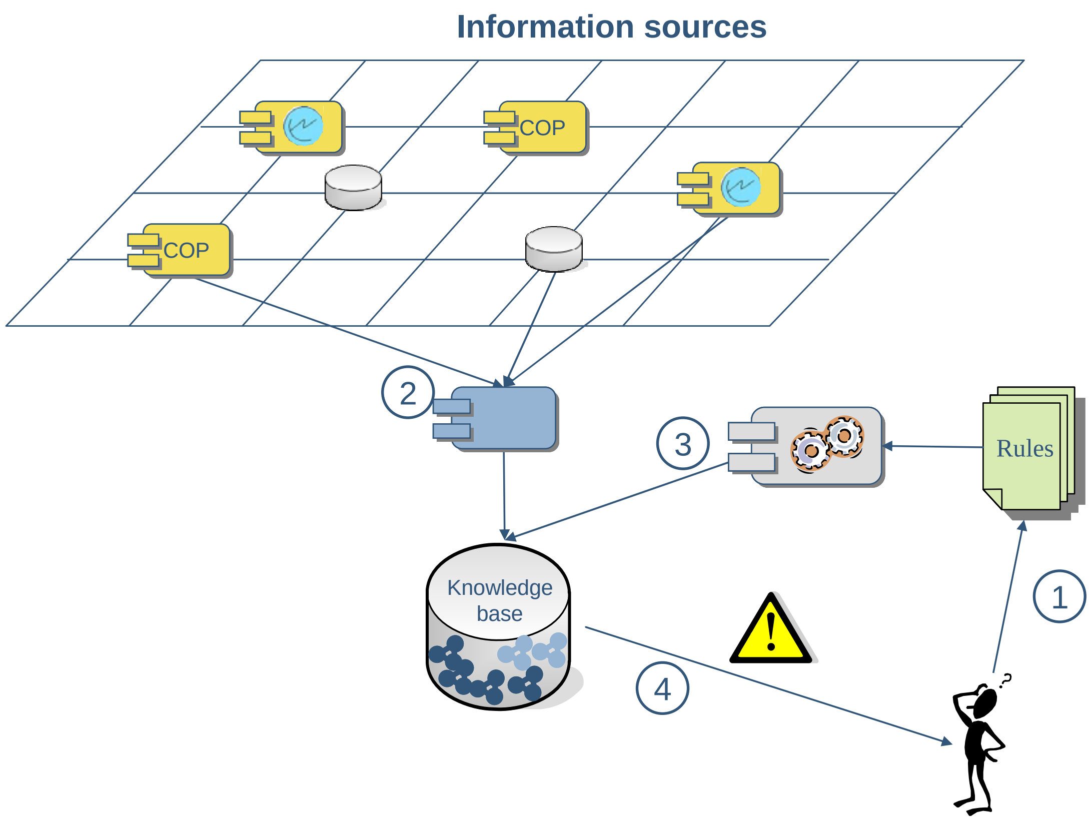


Figure 5.9 An example of semantic technologies used in information fusion

> In information fusion, ontology-based reasoning can assist in discovering non-observable relations between objects. The reasoning can be based on user-defined rules.

5.5 Semantic Service-Oriented Architecture (SOA) and Semantic Grid

There is a lot of effort going into the development of service-oriented architectures (SOAs) these days. SOA is an architectural paradigm that is based on loosely coupled services as building blocks and the orchestration of such services into composite services. The reason for doing this is that it becomes easier to change the orchestration as business needs change. Today, Web Services is the most popular technology for implementing a SOA, and there is a host of different specifications for creating the various parts of a SOA. To create a message-driven SOA, it is common to use an Enterprise Service Bus (ESB) as a SOA infrastructure. ESBs may provide various core services like reliable messaging, storage, transformation, message routing, management, and orchestration.

Whereas Web Services and ESBs provide interoperability plumbing, meaning that the *protocols* used for messaging are becoming standardised, there are still issues in the area of *data* interoperability. Web Services use XML Schema to restrict the syntax in the message payloads.

Since XML Schema has no scalable way to link between various schemas, we may quickly end up with the exponentially growing case illustrated in Figure 5.2 (a). By creating a semantic layer with ontologies that represent the different schemas, it becomes possible to map different source ontologies to domain ontologies. The domain ontology may in turn be used to mediate between different source ontologies, like in the linearly growing case in Figure 5.2 (b). The use of semantic technologies in ESB mediation services is shown in Figure 5.10. Here, an orchestration engine makes use of the semantic mediation service to combine information from command and control (C2), geography, and logistics services into a common operational picture (COP).

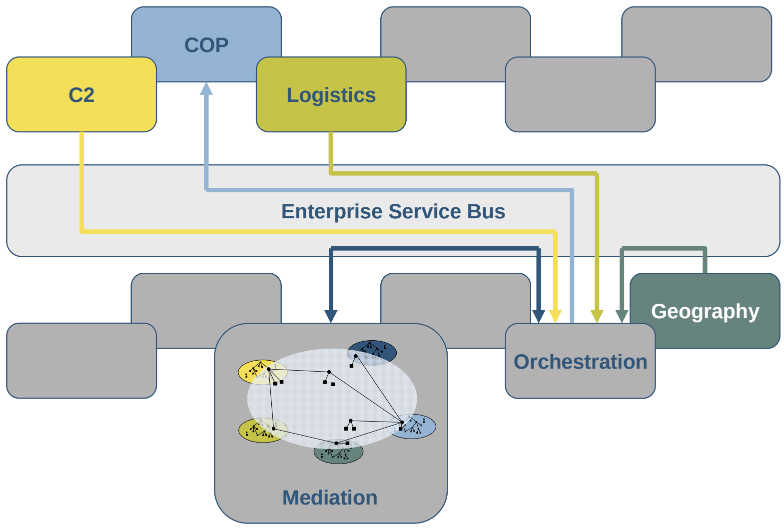


Figure 5.10 Semantic mediation as a core service in an ESB

Figure 5.11 shows the semantic SOA concept as a combination of several other technologies.

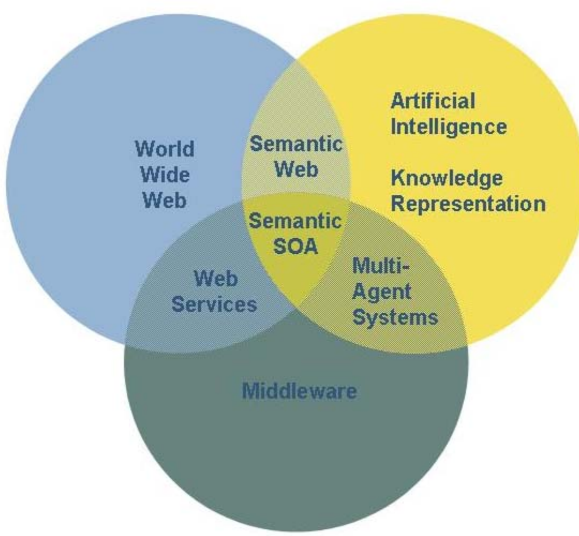


Figure 5.11 Semantic SOA shown as an intersection of Web Services, Multi-agent systems and the Semantic Web

Integration and mediation are probably the most immediate applications of semantic technologies in current Web Services-based SOAs, and as the number of schemas grows they are important functionalities. There are, however, several other areas where the use of semantic technologies is expected to enhance current SOA technologies, including discovery, policy description, data fusion, and planning. This allows creation of systems with very interesting capabilities, like self-healing and self-assembling systems and so on (GRID Today 2007).

As standards for Semantic Web Services mature, SOAs may benefit in terms of increased flexibility and dynamic service discovery, invocation, orchestration/composition, and negotiation between services and clients. As an example, Figure 5.12 shows how a user asks a question and how an agent handles discovery, composition and invocation of services on behalf of the user. This is actually the dynamic version of the ESB figure above in Figure 5.10. The user agent is represented as the blue component in the middle. Semantic service descriptions (SD) are searched by the agent to find services that (partially) match what the user has specified. When the services are found, a composed process combining different services is executed, and the services are invoked by the user agent. Finally, the user receives the result of the initial question it posed to the user agent.

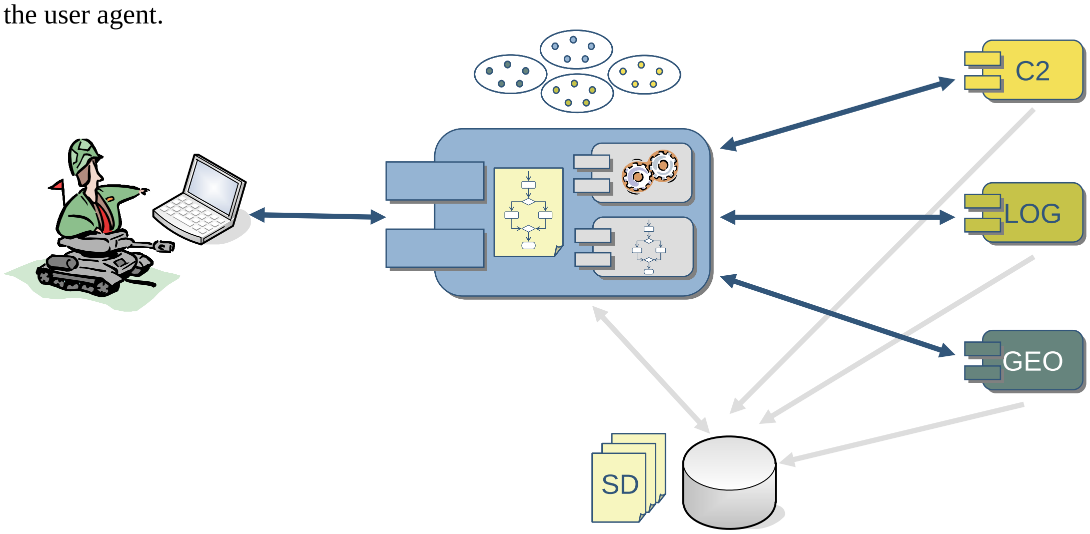


Figure 5.12 Discovery, composition and invocation of Semantic Web Services

Even further down the road, several research communities envision a semantic grid. The focus here is on grid capabilities like robustness, lifecycle management and negotiation combined with agent capabilities like resource allocation, workflow, learning, planning, interaction, and coordination (De Roure & Jennings 2005). This probably lies well into the future.

Using semantic technologies can allow for more flexible SOAs, especially in the area of mediation. At a later stage, automatic discovery, invocation, and composition of services can be made possible by semantic service descriptions.

5.6 Semantic Search

Searching has always been an important way to retrieve information from large information sets. This holds true whether the information set in question is the World Wide Web or a company's database systems. Traditionally, information retrieval using search is based almost purely upon text-matching techniques: A user enters a set of words that describes the desired information, and the result of the search is one or more documents where these words are present.

Semantic search, i.e. the use of semantic technologies in search, looks promising as a way to enhance traditional search strategies. The main contribution from semantic technologies in this regard is the handling of concepts and relations. Every concept is related to other concepts in various manners, and by utilising these relationships semantic search has the potential to provide search results with higher relevance to the user than traditional, text-based search alone.

At the heart of semantic search are the ontologies – the semantic models that provide a way to encode relations. In Figure 5.13 is shown an example of an ontology that will help us exemplify semantic search.

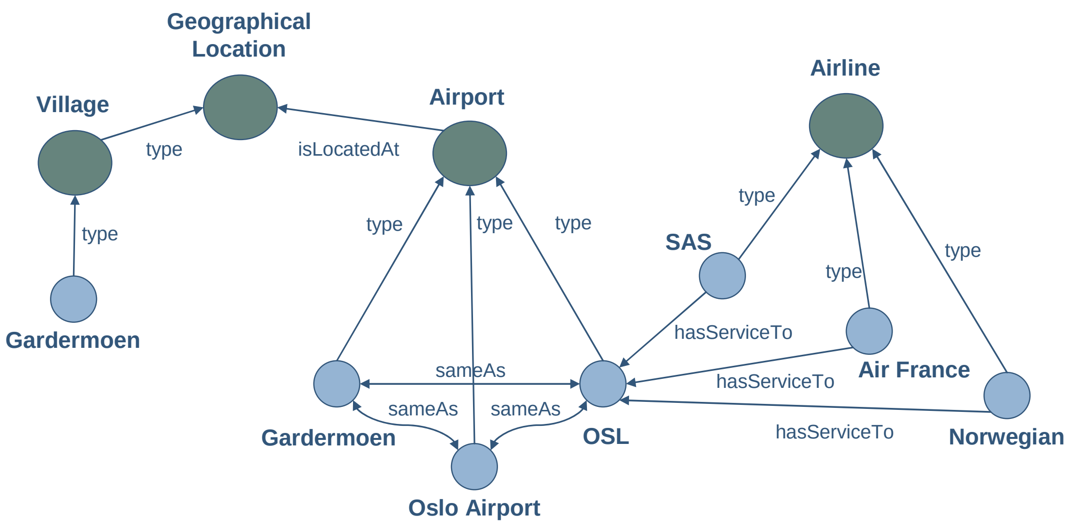


Figure 5.13 The airport example ontology with concepts (large green circles), individuals (small blue circles), and relations (links between circles)

For the example, let us begin with the simple search string 'airlines + OSL' where OSL is the airport code of Oslo Airport. Using the traditional search strategy, a search like this will normally return a lot of documents out of which only a few has any connection to the question this particular user wanted an answer to: A list of airlines with service to Oslo Airport. The reason for this is that the user has no way to encode the relation between the search terms: That the airlines of interest are airlines with a serviceTo relation with Oslo Airport. If these documents were encoded following an ontology like the one in Figure 5.13, the hasServiceTo relation would make sure this relation was understandable to a computer – in this case the search engine. Because of the importance of the relations, one can also find semantic search referred to as relation-based search.

Two important relations when it comes to search are the relations sameAs, which here is used to denote that two individuals represent the same object, and type, which denotes either a concept that is a specialisation of another concept or an item that is an individual of a concept type. As an example on the handling of the sameAs relation, say that the user searching for airlines with service to Oslo Airport this time enters ‘airlines + “Oslo airport”’ into a traditional search engine. Then documents with relevant information using synonyms for ‘Oslo airport’ (like ‘OSL’ or ‘Gardermoen’) will not be found unless the search engine somehow knows that the term ‘OSL’ has these two synonyms. This is usually not the case, unless the documents and search engine both adhere to an ontology which explicitly states this relation.

The type relation is also important in the search-related process of information navigation - the act of browsing through information to find what you are looking for. This can be thought of as the same kind of browsing as is done on the World Wide Web by following hyperlinks. The links will be what we here have referred to as relations, and one common way to browse information is to follow a link from a generic to a more specific concept. This also gives rise to one of the other names of semantic search: concept-based search. The topic of information navigation using semantic technologies is further covered in section 5.7.

Another important possible gain by using semantic search is that it can provide disambiguation. This is needed for handling homonyms (word with several meanings). A classical example of this is a search for ‘jaguar’. Does the user have in mind a new luxury car or a rather exotic pet? The use of an ontology provides a way to specify exactly what to search for. Related to our airport example, a search for Gardermoen will, for example, be ambiguous as the term Gardermoen is used for a Norwegian village as well as an airport, as seen in Figure 5.13. Using this ontology as a starting point, a user can specify that s/he is interested in the village Gardermoen and not the airport.

Semantic search exploits concepts and relations in semantic models to reduce ambiguity and give higher precision compared to traditional string-based search.

5.7 Information Presentation and Navigation

Given a situation of information overload, the task of finding or presenting relevant information is a major challenge. For the user, an important success criterion is to recognize the core concepts and relations of the information that is presented; i.e. the ontology that the information system builds upon.

These ontologies were previously implicit in the presentation tool, and might even be hard-coded by the programmers. Now there is a growing recognition of the value in creating explicit models on which to base the presentation. The value of the model-based presentation lies in facilitating the user’s navigation according to a “map” describing assumingly known concepts, individuals and their relations.

An example may clarify the meaning of this: Given a system containing a collection of news articles, and a small ontology describing the concepts Organisation, Person and InterestArea which are all interrelated. If each news article is tagged by ”author” (Person) and ”theme” (InterestArea), we can from a given article normally find other articles by this author and/or of this theme provided that these articles are tagged according to the same ontology.

Using the individuals of the ontology, the system will be able to dynamically create navigation links suggesting e.g.

- other related themes,
- other themes related to this author, and
- publishing organisations relevant to theme or author.

The benefits of the explicit model are twofold:

1. Clearly stated concepts that will ease the user’s understanding.
2. The ability to dynamically update the information content of the model (add new themes and authors, create new relations, and so on).

A more business-relevant example would be to base navigation on the model of an organisation’s departments and business areas, core products and services, localities, etc.

By navigating using to the ontology as the map, we exploit the fact that a piece of information may be annotated with reference to several independent concepts and individuals, and as such there will be a number of different ways to navigate in the direction of the desired result.

The ISO standard Topic Maps (TopicMaps.org 2007) is widely used to create a simple but powerful mapping of core concepts (topics) and their relations. Topic Maps are generally considered very user-friendly but they are too informal to be used as basis for formal reasoning. Similar systems based on RDF and OWL exist, described as doing faceted browsing (SWED 2007) or relational navigation (Siderean Software 2007)).

A popular web-tool for collaborative efforts and sharing of information is the type of systems called wiki. A wiki is a medium on the web which can be edited by anyone with access to it. It provides an easy method for linking together information provided by many different sources. Wikipedia (Wikipedia 2007) is an excellent example of this. A semantic wiki is a wiki that has an underlying model of the knowledge described within its pages. In its ideal form the semantic wiki would give us the ultimate tool where the information content and the describing ontology could be maintained in parallel. However, when implementing semantic wikis, experiences show that you can soon run into practical difficulties regarding expression complexity and unsolved limitations. Nevertheless, the vision of a semantic wiki has many benefits, and should be further explored.

Information navigation based on explicit models (ontologies) makes it easier for the user to find relevant information.

6 Military Applications of Semantic Technologies

The potential capabilities of semantic technologies have so far in this survey been considered in the context of general business. This section will focus on how the military domain may benefit from the potential of semantic technologies.

6.1 Network Based Defence

One of the primary goals of current military concept development in Norway is Network Based Defence (NBD), enabling units to collaborate across the traditional hierarchical organisation structure. The concept is based upon gaining information superiority that can be used to increase situational awareness, tempo in execution of command and operations, fighting capabilities, survival and self synchronisation. All this will in the end contribute to higher effectiveness. When it comes to information, the general idea is that all information already existing somewhere should be shared with anyone who needs it, provided there are no security restrictions involved.

The technological aspects of NBD have had a lot of the attention so far. Even though technology is an important enabler of these ideas, the most important changes will be those needed to adjust the organisation to utilise the new opportunities given by the technology. With a more dynamic and flexible organisation there will also be a need to share information in a more dynamic way, i.e. with partners that may not be known at system design time.

NBD is being developed in three consecutive phases: The first phase, Initial NBD, is where we are as of 2007. The NBD concept is spread throughout the organisation, and some ‘quick wins’ are implemented in the existing infrastructure. In the second phase, Integral NBD, there will be a need for considerable investments in new technology, and, most importantly, the organisation and doctrine have to be changed to adjust to the new concept. Phase three, which may be translated to Pervasive NBD, contains smaller changes, focusing on harvesting the benefits of the investments made in earlier phases, and to make further improvements and optimisation.

A technological premise for NBD is an infrastructure to enable widespread and flexible information exchange. This infrastructure is by the Norwegian Defence referred to as the information infrastructure (INI). The similar NATO term is Networking and Information Infrastructure (NII). An INI consisting of layers for communication, services and data integration, all supporting the collaborative processes of the enterprise is shown in Figure 6.1. The figure also shows how the syntactic and semantic levels build upon the technical level to complete the INI.

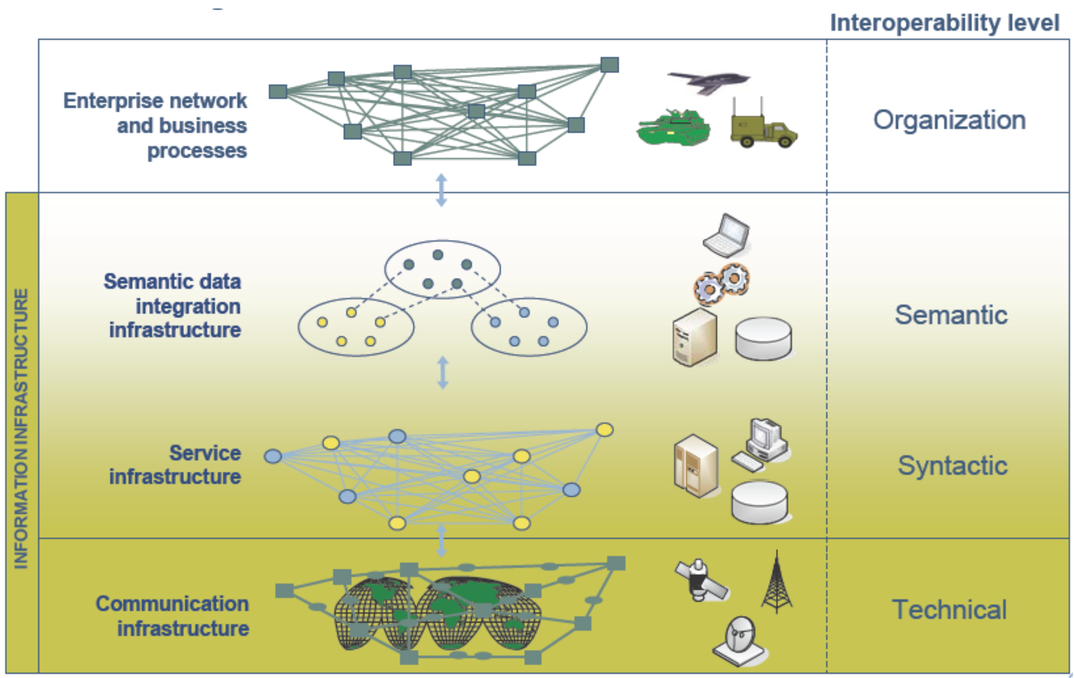


Figure 6.1 The Information Infrastructure in Network Based Defence

In the early stages of NBD, the main focus has been to move away from stove-piped systems to systems that are capable of data exchange using common formats. This is an important and necessary phase. However, we assume the implementation of NBD to bring new challenges:

Integration of information from semantically heterogeneous sources

Even though different systems get the ability to send data to each other, there is no guarantee that the data is neither syntactically nor semantically coherent. For a given system there will be a need to make mappings to convert external data to the system's own internal representation, hence converting it to meaningful information for the system. In the short term, this will probably be the widest use of semantic technologies, realising that most existing systems will not change their internal data representation, even though they are made able to communicate with other systems.

High demands on flexibility and dynamicity

Flexible and dynamic services will become a necessity in order to be able to interoperate with unanticipated services. In NBD, the traditional hierarchical approach will no longer be valid when it comes to the structure of information exchange. Even the organisational structures may become less hierarchical. Units will be grouped in the best way to fulfil the tasks, and this may change for every task. This means that it is no longer possible to tell in advance who you will cooperate with, thus new principles for the exchange of information are needed.

Information overload

From not getting very much information, this will soon change to too much information. Search engines have been available for quite a while, making important contributions to the process of extracting the relevant parts. But traditional text-based search has limitations. If you search for the term ‘Fridtjof Nansen’ in the military sense, for example, then you are probably interested in the frigate or the class of frigates and not the person. Traditional text-based search cannot differentiate these domains. In general, the number of hits gets so high that the relevant information gets hidden in between all the irrelevant. Even the amount of relevant information can get too big; there is also a need to know the quality of the relevant information (to find the most relevant of the relevant).

6.2 Applying semantic technologies to the military domain

Semantic technologies are trying to solve problems in a more automated way. Changes in business rules and system behaviour have traditionally been done by programmers, demanding time-consuming and labour-intensive efforts, both scarce resources in a live situation. Semantic technologies have potential for more dynamic solutions than the traditional hard-coded program logic, making it possible for system users to adjust to changing requirements during the course of action. In many cases this means that semantic technologies can solve problems that would otherwise not be dealt with.

An important application area for semantic technologies is decision support. Military intelligence is a prime contributor to the common operational picture (COP), which is the combined view of all known relevant information, especially regarding enemy forces, and is an important basis for decision support. Information may come from both national and allied military sources, as well as several civilian sources. There may be a need to integrate the information from all sources, as well as aggregating it to get the overview. The application of semantic technologies on these challenges is presented in sections 5.1 (information integration) and 5.4 (information fusion). The integration and aggregation may be done at several levels, as more information gets available at higher command levels. Some of the information may also be highly classified, and cannot be distributed to everyone. The goal is to produce an operational picture that can be distributed and utilised by all other forces.

Semantic technologies are about making the meaning of information explicit, so that a computer can assist in finding the relevant information. Information may come from many heterogeneous sources, where finding it is only one part of the puzzle, the trickier part may be to merge the information. It is not realistic to make all information sources use the same underlying structure (data model), especially since some of them may be outside of military control. Therefore it is important to also be able to extract information from sources created without using semantic technologies and merge it into a common picture.

The previously described need for flexible and dynamic information services is a primary basis for the use of service-oriented architectures (SOA). SOA implies that information is published without knowing all the receivers and the receiving part searches for information without knowing the provider. Both data providers and consumers may use a service registry to publish and find services. SOA is expected to be an important premise for the construction of the INI.

SOA is also one of the important technology recommendations in NATO Network Enabled

Capability Feasibility Study (NC3A 2005). Figure 6.2 shows their illustration of increasing levels of operational collaboration developing over time. Note that Semantic Web capabilities are depicted as the information integration service paradigm following SOA.

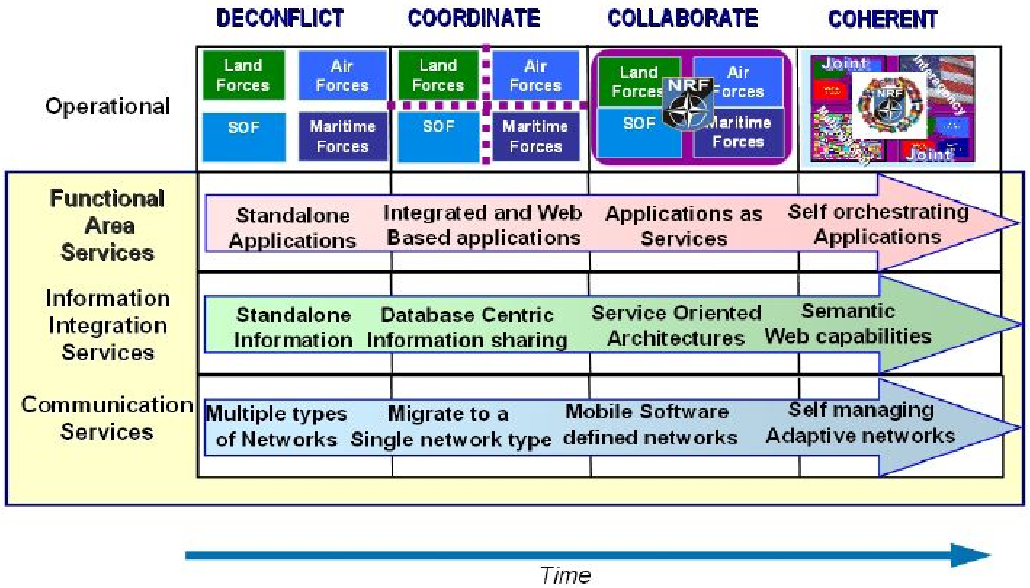


Figure 6.2 Evolving requirements and technology trends. From (NC3A 2005)

Semantic SOA is the result of applying semantic technologies to SOA, as described in section 5.5. Initially, we expect that core services in current SOAs like mediation, information integration and fusion, and federated querying will increasingly become ontology-driven. It is expected that the future will bring semantic service descriptions that can be input to automated reasoning processes that will improve the finding of the most applicable services (matchmaking). Semantic service descriptions also have the potential to enable dynamic service invocation and automated service composition. Future core services in the INI may benefit from these possibilities in order to further extend the flexible and dynamic properties that are very important.

We know that communities have different needs when it comes to modelling. As experienced by the Multilateral Interoperability Programme (MIP), it is not realistic to create one single data model that can capture all relevant information. MIP has standardised information for the operational domain, and aims to maintain interfaces to other domains' models. For more information on MIP, see (Gagnes et al. 2007). The approach of combining data models may probably benefit from the use of ontological tools, at least for the conceptual parts of the modelling. The computational nature of ontologies ought to have theoretical advantages to the traditional entity-relationship approach when it comes to automated support for the combination and linking of models.

Semantic technologies offer a wide range of possibilities, from relatively simple (but currently labour intensive) mappings between different models/ontologies, to reasoning on the available information to derive new information. The first part may be utilized in operational systems in the short term to reduce manual work, while the latter part probably needs some more research before it is fieldable.

The nature of the (Norwegian) Defence as a controlled environment where standards and regulations may be introduced in the information systems in an efficient manner is a potential key to achieving early benefits from emerging technology. Examples may be found in Chapter 5 and evolves from building ontological models that are the bases for applications like information navigation, automated integration and semantic SOA. To achieve results from semantic search, a controlled information sphere like (specific domains within) the Norwegian Defence has an excellent opportunity to create the necessary semantic annotations in a sufficiently large volume to be useful.

Achieving Network Based Defence (NBD) will require flexible and dynamic information services. Semantic technologies have the potential to contribute to fulfil this requirement. Decision support, intelligence and “INI as a Semantic SOA” are example areas where value may be added by using such technologies.

7 Selected Issues

We see that there are still several issues regarding semantic technologies. This is not very surprising considering the area in general is still rather young and immature. In this section we focus on a selection of the issues known to us: The possible use of ontologies to support model-driven development, XML Schema vs. Semantic Web technologies, the lack of convergence of the different Semantic Web Services initiatives in the near future, the ontology mapping problem, reasoning and decidability, and the maturity of the available semantic technology tools and infrastructure elements.

7.1 Model-driven development and the Ontology Definition Metamodel (ODM)

An important property of ontologies is their modelling abilities. They offer an expressiveness and flexible precision that deserves to be further exploited. An interesting area where these techniques could be applied is the field of model-driven systems engineering. The Object Management Group (OMG) has defined an approach for this called Model Driven Architecture (MDA) (Object Management Group 2007a).

The main idea of MDA is to separate business and application logic from the underlying computer platform technology. By building platform-independent models of a system’s behaviour, we create a foundation that enables the automatic generation of platform-specific models and finally executable code.

The MDA approach relies heavily on the existence of standards, methods and tools for modelling support and to generate the platform-specific model. The present lack of automatic tools available may be seen as a major objection to MDA. Nevertheless, MDA describes the ideal top-down approach for developing information systems.

An interesting question is whether MDA would benefit from starting with an ontology as the first and primary high-level model. The flexible precision of ontologies and their ability to perform formal reasoning might give important contributions to consistency checking throughout the development life cycle.

Ontologies are defined using different languages. Although OWL has become the Semantic Web standard, Topic Maps, UML, entity-relationship models, and so on, may be defined as ontological models as well. To overcome this diversity, OMG has defined the Ontology Definition Metamodel (ODM) (Object Management Group 2007b) that defines mappings between a number of these definition languages.

The model-driven approach recommends building systems at a higher level of abstraction to obtain a potential benefit of automatic generation of programming code to different platforms and sets of requirements. Use of formal ontologies may add value to the prescribed top-down processes. In theory these ontologies could be maintained and reused throughout the enterprise.

The issue here is whether ontology-based MDA can be done in practice. MDA has had slow progress and relatively low attention for quite some time. One of the critical points is MDA's dependency on supporting software and tools. When these aspects mature, the discipline of systems engineering may have substantial benefit from the modelling capabilities of ontologies.

7.2 XML Schema and Semantic Web Technologies

There are some points that are worth noting on the difference between XML Schema-based information exchange, and the Semantic Web family of technologies. As described in section 3.1, a distinguishing property of an XML document is that it represents a tree, whereas an RDF model represents a graph.

XML Schema-based information exchange, that is used with Web Services, is based on a schema that restricts the syntactic structure and content that an XML document can have if it is to be validated. This means that an XML Schema can be very specific about which elements can occur in a document, how many, and of which data type the different elements and attributes can be. One can also specify complex types, assembled from simple types. Further, one can define very strict syntactic types, e.g. valid licence plate number formats, social security number formats and so on.

We can say that Web Services technology is document-centric, whereas RDF and SPARQL are data-centric, and cannot be restricted in the same way. There is no way to specify exactly or limit what can be exchanged in the way that XML Schema allows. The roles of RDF Schema and OWL are different than that of XML Schema, dealing with the *meaning* and the relation between concepts of the data exchanged. XML Schema does not attempt to do this at all.

Of course, there are mechanisms that allow conversion between the two worlds, such as XSLT scripts. Another example is the Gloze contribution to the Jena toolkit (Battle 2006). Further, the next version of OWL (OWL 1.1) will allow properties to be of complex XML datatypes. This will eventually help bridge the two worlds.

7.3 Semantic Web Services Convergence

As presented in section 3.5, there have been, and still are several Semantic Web Services initiatives. Although the SAWSDL extension to WSDL has become a recommendation, it is a very limited subset of the issues Semantic Web Services researches initially set out to solve. Because of the lack of agreement, researchers working on the different initiatives are not pulling in the same direction, delaying the availability of development tools and execution environments that are feasible for production-quality systems. There are currently few, if any, commercial tool vendors that support this subset of semantic technologies.

We can only hope that in the future the more advanced areas of Semantic Web Services will converge into standards that eventually will be implemented by vendors (shown in Figure 7.1).

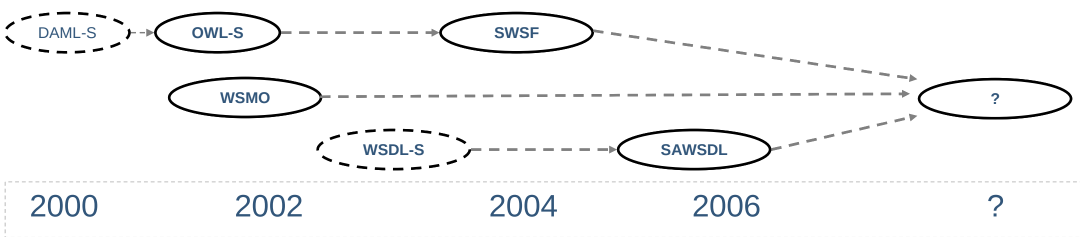


Figure 7.1 It is unclear when Semantic Web Service initiatives will converge

7.4 Ontology Mapping

One of the most important parts of semantic technologies is ontology mapping (Euzenat & Shvaiko 2007). This is especially true in the context of information integration, where ontologies may be auto-generated from source applications or systems and semantic mismatch between the ontologies needs to be resolved. We will look at two aspects of ontology mapping: Structure and mechanisms.

There are two main structures for mapping ontologies, although combinations of these may be used. The first one is the decentralised, or federated, structure where mappings are done in a point-to-point way between the ontologies. The other one is the centralised or hierarchical, where a mapping is made to an ontology on a higher level. This can result in a three or graph of ontologies. In (Kuehne 2006), a hierarchical structure is recommended for NATO.

As for different mechanisms that can be used to map ontologies, there are no existing best practices. We have identified three techniques to use; OWL statements, SPARQL CONSTRUCT queries, and rules. The three techniques may be used in combination, since they each have their own benefits. Nevertheless, in the following we describe them individually.

The most obvious mechanism for mapping (and extending/linking) ontologies is to use the constructs in OWL that support this. In OWL, it is possible to declare that a concept is equivalent to or a subclass of another concept. The same is possible with properties. Additionally, it is possible to declare that individuals are the same (as shown in section 5.6). The advantage of using OWL is that it is supported in standard reasoners. Further, since RDF/XML is the standard serialisation of an OWL ontology, a standardised way to exchange mappings exists. The drawback is that it seems that ontologies must be structurally compatible for OWL to be expressive enough for mapping purposes.

When two ontologies are structurally different, the SPARQL CONSTRUCT mechanism may be a better solution. This mechanism inserts query results into a template graph structure, and can therefore be used to convert instances and properties into new instances and properties in potentially quite different structures. As such, it may be perceived as a form of reasoning. The disadvantage of using this method is that it is not continuous, meaning that it will not execute until no further results have been inferred. This means that a CONSTRUCT query that is run twice will generate the same results twice, having the undesired effect of doubling all triples from the previous query run. Finally, there is no standardised way to exchange mappings, since no standard XML serialisation exists for SPARQL queries.

The final of the main ways to map between ontologies is using rules. Like SPARQL CONSTRUCT, using rules also needs its own execution, since few standard OWL reasoners currently also support rules. The main advantage of using rules, are so-called built-in functions, e.g. the product of two numbers, which often is necessary to map between ontologies. The drawback of using rules is again the serialisation bit, since none of the proposed rule formats have become W3C recommendations. However, the Rule Interchange Format (RIF) will probably take care of this in the future.

There exist solutions that combines several of the above mechanisms, see e.g. (Knublauch 2007). This is called hybrid reasoning.

It is worth mentioning that sometimes, because of the need for built-in functions, part of the mapping currently must be done at the syntactical level with current tools. For example, when one needs to split or concatenate strings, manipulate date and time formats and so on. Extensible stylesheet language transformation (XSLT) is a good candidate in this area. We expect to see tools combining several or all of the above mentioned mechanisms in the future.

7.5 Reasoning and Decidability

A knowledge representation language is decidable if there exists an algorithm with the ability to decide whether any valid statement in the language is true or not. Description logics-based languages, like for example OWL-DL, are known to be decidable (Horrocks, Patel-Schneider, &

van Harmelen 2003), but in (Horrocks, Patel-Schneider, & Bechhofer 2005) it is shown that extending a description logics-based language with Horn clause rules can lead to undecidability. I.e. there is a risk that a reasoner based on the extended language will not be able to tell whether a valid statement is true or not. Horn clause rules represent an attractive extension to OWL due to the several efficient inference procedures developed for them. An introduction of Horn clauses and their logical features can be found in (Hodges 1993).

As argued in section 3.3, however, it is often desirable to enhance knowledge description formalisms like OWL in order to obtain higher expressivity. This leads to a need for doing reasoning based on both OWL axioms and rules, so-called hybrid reasoning, and undecidability can become a problem. Undecidability in hybrid reasoning is a topic of current research, and among the suggested solutions to this problem is the concept of DL safe rules, introduced in (Motik & Sattler 2004).

7.6 Tool and Infrastructure Maturity

For semantic technologies to succeed, it is crucial that development tools and runtime infrastructures become available. We envision several kinds of technologies that may either appear as individual components or as parts of integrated development environments. Below, we briefly discuss tool and infrastructure support for each phase in a semantic technology project lifecycle.

Starting with the development phase, it will be important to have ontology development tools, as ontologies can become quite large and complex. Such tools may help users visualise, inspect, and refactor ontologies, check the consistency of their ontologies, like with e.g. the Protégé OWL tool (Protégé 2007), and assist users in mapping between different ontologies in various ways. Support for automatic generation of ontologies based on existing XML Schema, UML models, and database schema, like with D2RQ (Bizer & Seaborne 2004), will facilitate reuse of modelling efforts, and is already supported in commercial tools like TopBraid Composer (TopBraid 2007). The Eclipse IDE has a subproject called Eclipse Modelling Framework (EMF) Ontology Definition Metamodel (Eclipse 2007) that aims at implementing the Ontology Definition Metamodel as introduced in section 7.1, which will allow generation of OWL ontologies or UML diagrams based on the same metamodel. As for integration with today’s technologies, implementations that create RDF from both structured and unstructured sources (e.g. web pages and text documents) will be important. Whereas the former can be hand-coded, commercial tools are emerging that support doing the latter, e.g. Cypher (Monrai 2007). For Semantic Web Service development, there should be tools that allow users to annotate their service descriptions, and to compose and test services.

The next phase, the deployment or runtime phase, is when the developed artefacts are deployed in the IT infrastructure. Since with semantic technologies one should mostly work with models, the existence of generic infrastructure services, or semantic middleware, is crucial. Such services may be inference or reasoning services, RDF stores, execution environments e.g. for federated queries, translation, and mediation between different formats and languages. A variety of labels will likely be put on such semantics-based infrastructure services, for example ‘semantic information hub’, ‘semantic broker’ and so on. In the research community, work is being done on creating execution environments for hosting, invoking, and mediating between Semantic Web Services as well. See section 7.3 for more discussion of this.

Finally, we envision general-purpose tools for viewing and navigating data based on ontologies. This could be both portals and thick clients. The point is that it should be possible to view the relations between instances of different ontologies, possibly through different filters, or ‘lenses’. Ontologies could be used to define such filters. Examples of current technologies that touch this area are Siderean (Siderean Software 2007) and SWED (SWED 2007).

In conclusion, technology and infrastructure is maturing in some areas like for ontology development, whereas in other areas, like Semantic Web execution environments or ontology mapping tools, very few mature implementations exist. This may of course be related to the lack of standards to base implementations on. We currently know little about the performance and scalability of the various infrastructure components, but we are aware that in some cases, today’s reasoners need too much time to compute inferences. We expect this to be improved over the next few years.

8 Summary

Using the take-aways presented throughout this report, we summarise that:

Semantic technologies *are software technologies that exploit the meaning of the information at hand and involve the use of an explicit knowledge model.* ***Compared to traditional information technologies****, semantic technologies offer tools to ease the making of more adaptable and flexible software.*

In the FFI project Semantini, Semantic services in INI (INformation Infrastructure), these technologies are studied in order to evaluate if, and how, they can be utilised in military applications. Of special interest is their use in providing the information infrastructure of the Norwegian military concept Network Based Defence (NBD) with flexible core services:

Achieving ***Network Based Defence (NBD)*** *will require flexible and dynamic information services. Semantic technologies have the potential to contribute to fulfil these requirements. Decision support, intelligence and “INI as a Semantic SOA” are example areas where value may be added by using such technologies.*

One of the strongest drivers for semantic technologies at the moment is the work related to the Semantic Web. The Semantic Web is a vision presented in (Berners-Lee, Hendler, & Lassila 2001), and can be summarised as:

The Semantic Web *is an extension of the current web where the use of knowledge representation techniques is envisioned to allow computers to collect data on the web on behalf of humans.*

Semantic technologies are in particular expected to do two things for the Semantic Web: (1) Make more data currently connected to the web globally available and (2) make services connected to the web more findable to users and other services by tagging them with metadata.

Semantic technologies are still a relatively young technology family, and as such there exists no consensus on what constitutes the fundamental technologies. Still, we have in this report singled out five technologies, or technology areas, which we feel represent the core of semantic technologies:

Knowledge representation *aims at representing and storing knowledge, making the knowledge accessible to computers. One of the most important models for representing knowledge is the graph based Resource Description Framework (RDF).*

Ontologies *are formal semantic models of a domain of interest. Ontologies can be linkable, and OWL is the DL-based web ontology language of W3C.*

Reasoning *in connection with semantic technologies provides the users with ontology design assistance and gives the capability of inferring new information from a knowledge base.In order to increase the restricted expressivity of ontology specification languages, there is a need for user-defined rules.*

It is possible to ***query*** *knowledge bases in a way similar to querying databases. SPARQL is the query language and protocol for querying RDF.*

Agents *are software components able to take advantage of computer-readable elements in information and services.* ***Semantic Web Services*** *are Web Services described using ontologies, facilitating automation of several tasks.*

Even though several fields of semantic technologies are still experimental and immature, some areas stand out as more mature. Examples include RDF and OWL, which after several years of use in the Semantic Web community is considered to be stable and mature. It is also interesting to note that lately there has been an increase in semantic technology-based products brought to the market.

The list of proven and (mostly) expected capabilities of semantic technologies has grown to become quite long, but we have chosen to focus on a selection of these capabilities:

Semantic ***information integration*** *provides a more scalable way to integrate information from heterogeneous sources, based on automatic reasoning and mapping between ontologies.*

Semantically enabled ***federated querying*** *utilises semantic information integration to query databases in potentially different organisations while preserving database autonomy.*

Semantic ***meditation*** *uses semantic technologies to resolve semantic and/or protocol mismatch between clients and services, possibly in a transparent way.*

In ***information fusion****, ontology-based reasoning can assist in discovering non-observable* *relations between objects. The reasoning can based on user-defined rules.*

Using ***semantic*** *technologies can allow for more flexible* ***SOAs****, especially in the area of mediation. At a later stage, automatic discovery, invocation, and composition of services can be made possible by semantic service descriptions.*

Semantic search *exploits concepts and relations in semantic models to reduce ambiguity and give higher precision compared to traditional string-based search.*

Information navigation *based on explicit models (ontologies) makes it easier for the user to find relevant information.*

We see that there are still several issues regarding semantic technologies. This is not very surprising considering the area in general is still rather young and immature. In this document we have focused on a selection of the issues known to us: The possible use of ontologies to support model-driven development, XML Schema vs. Semantic Web technologies, the lack of convergence of the different Semantic Web Services initiatives in the near future, the ontology mapping problem, reasoning and decidability, and the maturity of the available semantic technology tools and infrastructure elements.

9 Conclusion

Semantic technologies are a promising family of information technologies that potentially can deliver critical capabilities also in the military domain. There is obvious potential in using a tool like ontology linking as a way to manage information models in large organisations like the Norwegian Defence, NATO etc. Another important point to note is that using semantic technologies for several capabilities facilitates reuse of ontologies and mappings. An example of this would be that the same ontology that is used to map against in an information integration scenario could be used to drive a relational navigation-based portal. As such, ontologies rather than systems may become the assets in the future information infrastructure. Some semantic technologies, like for example RDF and OWL, enjoy a relatively high level of maturity, and lately there has been an increase in semantic technology-based products brought to the market. However, as several members of the semantic technology family still are experimental, more research and experimentation is needed to evaluate whether they can deliver as promised and hoped.

It is likely that several of the core services in the future information infrastructure (INI) will benefit from semantic technologies. In order to further evaluate this technology family for use in the Norwegian Defence, there is a need for more thorough studies in cooperation with military domain experts.

Appendix A Abbreviations

ABox	Assertion Box
AI	Artificial Intelligence
BPEL	Business Process Execution Language
BPM	Business Process Modelling
C2	Command and Control
COP	Common Operational Picture
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DL	Description Logics
EMF	Eclipse Modelling Framework
ESB	Enterprise Service Bus
FFI	Forsvarets Forskningsinstitutt (Norwegian Defence Research Establishment)
GRDDL	Gleaning Resource Descriptions from Dialects of Languages
HTML	HyperText Markup Language
IDE	Integrated Development Environment
INI	INformation Infrastructure
JDL	Joint Directors of Laboratories
JSON	JavaScript Object Notation
MDA	Model-Driven Architecture
MIP	Multilateral Interoperability Programme
NATO	North Atlantic Treaty Organisation
NBD	Network Based Defence
NII	Networking and Information Infrastructure
ODM	Ontology Definition Metamodel
OMG	Object Management Group
OWL	Web Ontology Language
OWL-S	Web Ontology Language for Services
PSL	Process Specification Language
RDF	Resource Description Framework
RDF-S	Resource Description Framework – Schema
RIF	Rules Interchange Format
RuleML	Rule Markup Language
SAWSDL	Semantic Annotations for WSDL
SOA	Service-Oriented Architecture
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language
SWRL	Semantic Web Rules Language
SWSF	Semantic Web Services Framework
SWSL	Semantic Web Services Language
TBox	Terminology Box

UML	Unified Modelling Language
W3C	World Wide Web Consortium
WSDL	Web Service Definition Language
WSMO	Web Service Modelling Ontology
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformation

References

Alberts, D. S., Garstka, J., & Stein, F. P. 1999, *Network Centric Warfare: Developing and Leveraging Information Superiority* CCRP Publications Distribution Center.

Baader, F., Calvanese, D., McGuinnes, D., & Nardi, D. 2003, *The description logic handbook* Cambridge University Press.

Battle, S. 2006, "Gloze: XML to RDF and back again", *2006 Jena User Conference*.

Berners-Lee, T., Hendler, J., & Lassila, O. 2001, "The Semantic Web", *Scientific American*, vol. 284, no. 5, pp. 28-37.

Bizer, C. & Seaborne, A. 2004. "D2RQ-Treating Non-RDF Databases as Virtual RDF Graphs". Proceedings of the 3rd International Semantic Web Conference (ISWC2004) .

Boley, H. & Tabet, S. 2001. "Design Rationale of RuleML: A Markup Language for Semantic Web Rules". Proc. Semantic Web Working Symposium (SWWS'01).

Davis, M. 2004, *The Business Value of Semantic Technologies*, TopQuadrant.

Davis, M. 2006, *Semantic Wave 2006 Part 1: Executive Guide to Billion Dollar Markets*.

De Roure, D. & Jennings, N. R. 2005, "The Semantic Grid: Past, present, and future", *Proceedings of the Ieee*, vol. 93, no. 3, pp. 669-681.

del.icio.us. 2007. "del.icio.us - social bookmarking". http://del.icio.us/ .

Eclipse. 2007. "Eclipse EODM". http://www.eclipse.org/modeling/mdt/?project=eodm .

Endsley, M. R. & Garland, D. J. 2000, *Situation Awareness Analysis and Measurement* Lawrence Erlbaum Associates.

Euzenat, J. & Shvaiko, P. 2007, *Ontology matching* Springer-Verlag, Heidelberg (DE).

Flickr. 2007. "Flickr - Photo sharing". http://flickr.com/ .

Gagnes, T., Eggen, A., Hedenstad, O.-E., Rasmussen, R., & Sletten, G. 2007, *Operative beslutningsstøtte-tjenester -- Fremtid NbF* FFI/RAPPORT 2005/03584.

GRID Today. 2007. "SAIC Building Semantic SOA". http://www.gridtoday.com/grid/1533211.html .

Haarslev, V. & Möller, R. 2001. "Description of the RACER system and its applications". 132-142. Proceedings International Workshop on Description Logics (DL-2001).

Hall, D. L. & McMullen, S. A. H. 2004, *Mathematical Techniques in Multisensor Data Fusion* Artech House.

Hodges, W. 1993, "Logical Features of Horn Clauses," in *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. M. Gabbay, C. J. Hogger, & J. A. Robinson, eds., Oxford University Press, USA.

Horrocks, I., Patel-Schneider, P. F., & Bechhofer, S. 2005, "OWL rules: A proposal and prototype implementation", *Journal of Web Semantics*, vol. 3, no. 1, pp. 23-40.

Horrocks, I., Patel-Schneider, P. F., & van Harmelen, F. 2003, "From SHIQ and RDF to OWL: The making of a web ontology language", *Journal of Web Semantics*, vol. 1, no. 1, pp. 7-26.

Jurafsky, D. & Martin, J. H. 2000, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* MIT Press.

Knublauch, H. 2007. "Hybrid Inferencing with TopBraid Composer 2.0". http://composing-the-semantic-web.blogspot.com/2007/02/hybrid-inferencing-with-topbraid.html .

Kuehne, S. 2006. "Ontology Layers for NATO-wide Knowledge Representation, NC3A Technical Note 1240 (Draft)".

McComb, D. 2005, *The CIO's Guide to Semantics*.

McGuinness, D. L. 2003, "Ontologies Come of Age," in *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, D. Fensel, J. Hendler, & H. &. W. W. Lieberman, eds., MIT Press, pp. 171-192.

microformats.org. 2007a. "hCalendar Microformat". http://microformats.org/wiki/hcalendar .

microformats.org. 2007b. "hCard Microformat". http://microformats.org/wiki/hcard .

Monrai. 2007. "Cypher - Natural Language Processing for the Semantic Web". www.monrai.com .

Motik, B. & Sattler, U. 2006. "A comparison of reasoning techniques for querying large description logic aboxes". Proc. of the 13th Int. Conf. on Logic for Programming Artificial Intelligence and Reasoning, Phnom Penh, Cambodia.

Motik, B. & Sattler, U. 2004. "Query Answering for OWL-DL with Rules". 549-563. Springer. Proc. of ISWC 2004.

NC3A. 2005. "NATO Network Enabled Capability Feasibility Study Volume 1: Overview of NATO Network-Centric Operational Needs and Implications for the Development of Net-Centric Solutions. Version 2.0".

Norheim, D. 2007. "Slide from a 'Semantic Interoperability' presentation". Computas AS.

Object Management Group. 2007a. "OMG Model Driven Architecture". www.omg.org/mda .

Object Management Group. 2007b. "Ontology Definition Metamodel". http://www.omg.org/docs/ad/06-05-01.pdf .

Polikoff, I. & Allemang, D. 2004, *Semantic Technology*.

Pollock, J. T. & Hodgson, R. 2004, *Adaptive Information. Improving Business Through Semantic Interoperability, Grid Computing, and Enterprise Integration* John Wiley & Sons.

Protégé. 2007. "Protégé OWL". http://protege.stanford.edu/overview/protege-owl.html .

Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lare, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., & Fensel, D. 2005, "Web Service Modeling Ontology", *Applied Ontology*, vol. 1, no. 1, pp. 77-106.

Sheth, A. & Ramakrishnan, C. 2003, "Semantic (Web) Technology in Action: Ontology Driven Information Systems for Search, Integration, and Analysis," in *IEEE Data Engineering Bulletin*.

Siderean Software. 2007. "Siderean Navigator". http://www.siderean.com/ .

Sirin, E., Parsia, B., Grau, B. C., & Kalyanpur, A. &. K. Y. 2007, "Pellet: A Practical OWL-DL Reasoner", *Journal of Web Semantics*, vol. 5, no. 2, pp. 51-53.

Steinberg, A. N. & Bowman, C. L. 2001, "Revisions to the JDL Data Fusion Model," in *Handbook of Multisensor Data Fusion*, D. L. Hall, ed., CRC Press, pp. 2-1.

Stephens, S., LaVigna, D., DiLascio, M., & Luciano, J. 2006, "Aggregation of bioinformatics data using Semantic Web technology", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, no. 3, pp. 216-221.

Studer, R., Banjamins, V. R., & Fensel, D. 1998, "Knowledge Engineering: Principles and Methods", *Data & Knowledge Engineering*, vol. 25, no. 1-2, pp. 161-197.

SWED. 2007. "SWED". http://www.swed.org.uk/swed/index.html .

TopBraid. 2007. "TopBraid Composer". www.topbraidcomposer.com .

TopicMaps.org. 2007. "TopicMaps.org". http://www.topicmaps.org/ .

Tsarkov, D. & Horrocks, I. 2006. "FaCT++ description logic reasoner: System description". 4130, 292-297. Springer. Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006) Lecture Notes in Artificial Intelligence.

W3C. 2004a. "OWL-S: Semantic Markup for Web Services". http://www.w3.org/Submission/OWL-S/ .

W3C. 2004b. "RDF Primer".

W3C. 2004c. "RDF Vocabulary Description Language 1.0: RDF Schema". http://www.w3.org/TR/rdf-schema/ .

W3C. 2005a. "Semantic Web Services Framework (SWSF) Overview". http://www.w3.org/Submission/SWSF/ .

W3C. 2005b. "Semantic Web Services Language (SWSL)". http://www.w3.org/Submission/SWSF-SWSL/ .

W3C. 2007a. "Gleaning Resource Descriptions from Dialects of Languages (GRDDL)". http://www.w3.org/TR/grddl/ .

W3C. 2007b. "RDFa Primer". http://www.w3.org/TR/xhtml-rdfa-primer/ .

W3C. 2007c. "Rules Interchange Format". http://www.w3.org/2005/rules/ .

W3C. 2007d. "Semantic Annotations for WSDL and XML Schema". http://www.w3.org/TR/sawsdl/ .

W3C. 2007e. "SPARQL Query Language for RDF". http://www.w3.org/TR/rdf-sparql-query/ .

W3C. 2007f. "Web Ontology Language (OWL)". http://www.w3.org/2004/OWL/ .

Wald, L. 1999, "Some Terms of Reference in Data Fusion", *IEEE Transactions on geoscience and remote sensing,* vol. 37, no. 3, pp. 1190-1193.

Wells, E. 2005. "Net-Centric Operations & Warfare (NCOW) Reference Model (Version 1.1) & NC3A Impacts". NATO Open Systems Working Group.

Wikipedia. 2007. "Wikipedia". en.wikipedia.org .