Integration of Semantic Resources and Tools for Business Intelligence

Thierry Declerck[1], Hans-Ulrich Krieger[1], Bernd Kiefer[1],
Marcus Spies[2], Christian Leibold[2]

[1]DFKI GmbH, Language Technology Lab, Stuhlsatzenhausweg 3,
D-66123 Saarbruecken, Germany
{declerck, krieger,kiefer}@dfki.de
[2]DERI, Universität Innsbruck, Technikerstraße 21a,
A - 6020 Innsbruck, Austria
{marcus.spies, christian.leibold}@deri.at

Abstract. We describe in this position paper the actual state of development of semantic resources, including a temporal ontology, and technologies in the context of a European R&D project dealing with Business Intelligence. We describe in some details the actual state of ontology development for guiding information extraction onto an ontology population task. We also present our actual efforts for implementing efficient reasoning platform acting over the knowledge base.

Keywords: Extracting (business) semantics from structured and unstructured documents, ontology languages, reasoning.

1 Introduction

MUSING is an R&D European project[1] dedicated to the development of a new generation of Business Intelligence (BI) tools and modules founded on semantic-based knowledge and content systems. MUSING integrates Semantic Web and Human Language technologies and combines declarative rule-based methods and statistical approaches for enhancing the technological foundations of knowledge acquisition and reasoning in BI applications. The impact of MUSING on semantic-based BI is being measured in three strategic, vertical domains:

- **Financial Risk Management (FRM),** providing services for the supply of information to build a creditworthiness profile of a subject -- from the collection and extraction of data from public and private sources up to the enrichment of these data with indices, scores and ratings;
- **Internationalization (INT)**, providing an innovative platform, which an enterprise may use to support foreign market access and to benefit from resources originating in other markets;

[1] See www.musing.eu for more details.

- **IT Operational Risk & Business Continuity (ITOpR),** providing services to assess IT operational risks that are central for Financial Institutions -- as a consequence of the Basel-II Accord – and to asses risks arising specifically from enterprise's IT systems -- such as software, hardware, telecommunications, or utility outage/disruption.

Across those vertical streams of MUSING, there are some common tasks, like the one consisting in extracting relevant information from annual reports of companies and to map this information into XBRL (Extended Business Reporting Language). XBRL is a standardized way of encoding financial information of companies, but also the management structure, location, number of employees, etc. (see www.xbrl.org). This is basically "quantitative" information, which is typically encoded in financial tables.

But for many Business Intelligence applications, there is also a need to consider "qualitative" information, which is mostly delivered in the form of free text in the annexes to the balance sheets in annual reports or in news articles. The problem is therefore how to optimally integrate quantitative information from the periodic reports and the day to day information provided by specialized news agencies. So for example imagine that you have a balance sheet from the company "Daimler-Chrysler" for the year 2006, but since September 2007, the company has been renamed "Daimler", and the press agencies will mostly only use this naming for reporting on the company. How can we deal with this, in order to take into account that the information on "Daimler" is closely related with the information on "Daimler-Chrysler" for the time till September 2007? We need here accurate information extraction (IE) systems, that detect in the news this change of name of the company, and which still allow for populating the MUSING ontologies, when only the new name of the company is being used by the reporting documents.

Work on IE and ontology population in MUSING is depending on temporal information associated with the document. So for example the date of publication of an annual report doesn't coincide with the end of the reporting period, and we have to extract the values for the starting and the ending time of the reporting period from the document itself. This information is typically included in the financial tables. The temporal information associated with certain quantitative information contained in those tables can be of two types: duration or instant (for example the number of employees given is valid for a specific instant in time, whereas the growth of certain financial indicators is valid for a certain period). This distinction has to be made explicit in our semantic representation of the relevant information in MUSING.

The name of the CEO is valid for the instant of time, which is the end of the reporting period. But very often the annex to the balance sheets is giving more detailed temporal information, and in case the company has had a change of CEO during the reporting period, the precise time in which this change has occurred is explicitly mentioned in the annex to the balance sheet, which is in a free text form. This information has to be extracted by the IE component of MUSING and used for populating the ontologies. But in this case we have a property in the ontology (the CEO relationship), with more than one valid value for the reporting period. We have to be able to cope with this fact in the instantiation of ontology classes in MUSING.

As a summary of our needs with respect to temporal information in the concrete task described above, we learned that we can not work with only synchronic relationships.

2 Integrated Ontologies in MUSING

In order to maximize the exploitation of past experiences on the one hand and to minimize duplication of effort on the other hand, we have searched for an ontological framework that would suit our needs and meet our qualitative requirements. We evaluated various frameworks such as PROTON (http://proton.semanticweb.org) and DOLCE (http://www.loa-cnr.it/DOLCE.html) thoroughly and were finally convinced that building our solutions on PROTON which was developed within the scope of the SEKT project (http://www.sekt-project.com/) would be of considerable value.[2] Furthermore, many general concepts such as Person and Company are already defined and can be used. We then either extended those concepts by sub classing or just added the - from our perspective - missing properties.

The latest original PROTON ontology (files protons, protont, protonu, protonkm; see http://proton.semanticweb.org/) from April 2005 has been slightly modified to integrate our treatment of time. This version is schematically shown in figure 1:

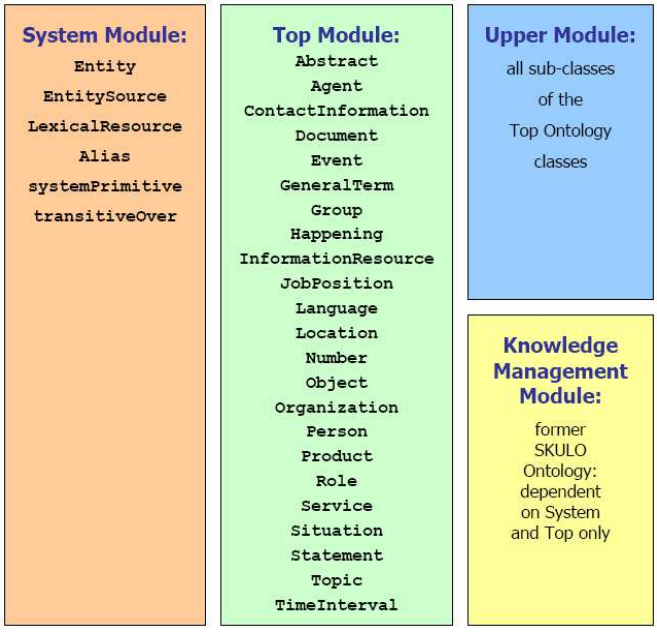

Figure 1. Overview of PROTON Modules (Terziev et al. 2005).

"The System module of PROTON, http://proton.semanticweb.org/2005/04/protons, provides a sort of high-level system- or meta-primitives, which are likely to be accepted and even hard-coded in tools that may use PROTON. It is the only

[2] However, PROTON is not a normative framework for the MUSING ontologies. Other ontologies such as DOLCE and LKIF (http://www.estrellaproject.org/lkif-core/) are consistently evaluated.

component in PROTON that is not to be changed for the purposes of ontology extension." (Terziev et al. 2005)

The contained Top-Level classes, http://proton.semanticweb.org/2005/04/protons, represent the most common definition of world knowledge concepts. These can directly be used for knowledge discovery, metadata generation and to interface intelligent knowledge access tools (Terziev et al. 2005). In MUSING the only top-level classes carried are Abstract, Happening, and Object.

The PROTON upper module, http://proton.semanticweb.org/2005/04/protonu, adds sub-classes and properties to the Top-module super classes to the concepts other than "Abstract, Happening and Object" from the original PROTON Top ontology.

And the PROTON Knowledge Management module: http://proton.semanticweb.org/2005/04/protonkm, which contains amongst others *InformationSpace* (collection of information resources), *User* (including his interests, stored in *Profile).* Further *LexicalRessource* provides with *Mention* a specialisation that can be used in the process of annotation.

. Basically, we have three main types of ontologies in MUSING:

(i) A modified version of the PROTON ontology, together with several large extension for the three domain of application we mentioned in the introduction,

(ii) A time ontology, and

(iii) Domain ontologies, like for example a OWL ontology encoding the information included in the German XBRL taxonomy

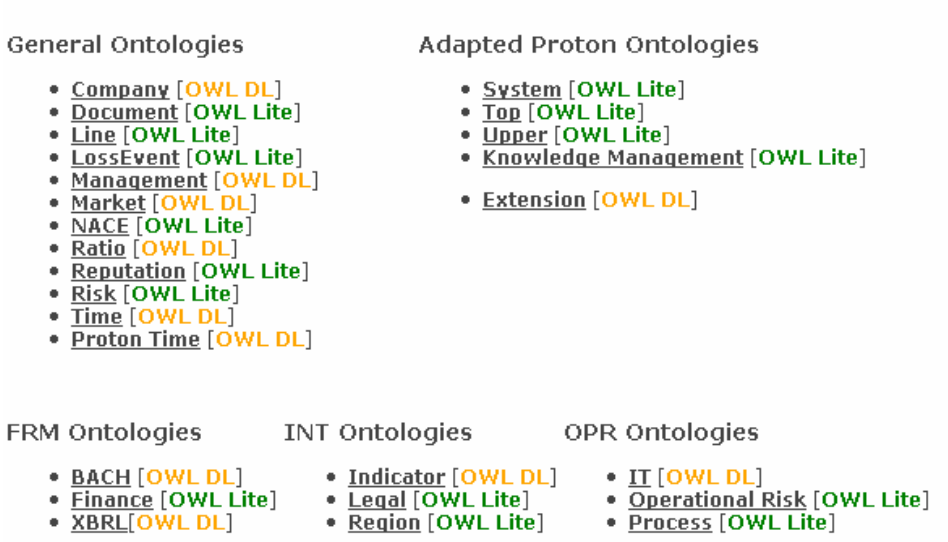

Figure 2: The organization of the ontology modules in the MUSING framework

The changes in the PROTON ontologies generated by MUSING needs affect the base URI of each ontology, changing from http://proton.semanticweb.org/2005/04/* to http://musing.deri.at/ontologies/v0.6/proton/*.

In the time ontology of MUSING, temporally-enriched facts are represented *through time slices*, four dimensional slices of what Sider (1997) calls a *space-time worm* (we only focus on the temporal dimension in MUSING). These worms, often referred to as *perdurants*, are the objects we are talking about. For instance, *Jürgen Schrempp* (JS) is a perdurant that comes up with several time slices, talking about his CEOship with Daimler Chrysler (DC), his resignation as CEO of DC, his membership within the supervisory board of Allianz and Vodafone, etc. All facts are associated with a temporal dimension, even if they are instants, i.e., having an infinitely-small extension.

Through the choice of taking PROTON as the base ontology, we are committed to the use of OWL as the representational language. Unfortunately, binary OWL properties cannot be easily extended by further time arguments. However, one can wrap property values plus temporal information in a time slice object. What had originally been an entity thus now becomes a time slice. The access to the time slices of a perdurant is handled via the hasTimeSlice property.

Although there are a number of very good reasons to favor OWL (like interoperability), we note here that the binarization of properties might have drastic effects on temporal representation (more space) and reasoning (more time).

What was originally an entity in PROTON has become a time slice in PROTON+Time. In order to keep changes small, we do not reduplicate the psys:Entity class hierarchy on the perdurant side. So, for instance, ptop:Person now represents a *time slice* of a *perdurant* that *acts* as a *person*. The general strategy is to move time-varying information into a perdurant's time slice and to move temporal-constant information to the perdurant itself. Thus a perdurant might have time slices of different types. For instance, perdurant SRI acts sometimes as an AcademicInstitution, but sometimes also as a Company.

The species of the model of the PROTON Upper module as it is currently available at http://proton.semanticweb.org/2005/04/protonu is OWL Full. The MUSING version available at http://musing.deri.at/ontologies/v0.5/proton/protonu contains mostly the same information as the original one but is slightly changed to fulfil the OWL Lite criteria. The evaluation of the species was performed with the most recent version of the Eclipse-based ontology editor TopBraid Composer.

Besides the time ontologies, there are currently ten ontologies, which are not assigned to any particular application. They cover the following areas: company, document, business line, loss event, management, market, ratio, reputation and risk. In addition, one covers the European industry standard classification system Nomenclature Generale des Activites Economiques dans L'Union Europeenne (NACE), which is equivalent to the Standard Industrial Classification (SIC) and the North American Industry Classification System (NAICS).

The set of Financial Risk Management (FRM) ontologies for MUSING currently consists of three ontologies. They cover (i) the structure for balance sheets, profit and loss accounts, statements of investments and statements of depreciation as used in the Bank for the Accounts of Companies Harmonised (BACH) database, (ii) the Extensible Business Reporting Language (XBRL) governed by XBRL International Incorporated, and (iii) a collection of concepts common in the domain. It can already be stated that the *Company* class of the PROTON Upper module will be the key concept with respect to time in the FRM applications in MUSING

The set of ontologies for the Internationalization applications in MUSING also contains three ontologies. The most important ontology is the one defining the indicators used to measure properties of political regions. It is based on a list of 162 indicators grouped into 14 categories. While one of the remaining ontologies in those applications covers the 28 provinces of India, the other one contains legal concepts with regard to internationalization.

In the ItOperational Risk applications of MUSING, we also introduce three ontologies. While two of the ontologies deal with processes and IT infrastructure, respectively, the remaining one describes operational risk in general and IT operation risk in particular. Relevant dimensions are area, cause and domain in the case of operational risk and in addition category for IT operational risk.

3 An Integrated Reasoning platform

Beyond, or on the top of OWL, we need a rule language to formulate native rule knowledge. Such a language should be able to take OWL constructs as basic building blocks into account. There have been several such proposals, most notably *SWRL*, the Semantic Web Rule Language (Horrocks et al. 2004).

Two freely avalable reasoners are much in the spirit of SWRL, viz., *OWLIM* (Kiryakov 2006; http://www.ontotext.com/owlim/) and *Jena* (Reynolds 2006; http://jena.sourceforge.net/), which we will present in a moment.

Overall, there are not many other (partial) implementations of SWRLish (sub-)languages[3]:

- The latest version of *Pellet* (Kolovski et al. 2006; http://pellet.owldl.com/) which we will use for initial TBox
- The latest version of Racer (Haarslev & Möller 2001; http://www.racer-systems.com/). But we were not able to activate the rule engine with RacerPro v1.9 (December 8, 2005) in a test setting with the a larger ontology (approx. 1,000 classes, 20,000 instances), since inital TBox and ABox consistency checking required more than 1 GB main memory.
- Jess, "the Rule Engine for the Java Platform" (http://herzberg.ca.sandia.gov/jess/) is a general rule engine, developed at the Sandia National Lab by Ernest Friedman-Hill. Jess does not provide a native built-in OWL support, but via the Protégé's SWRL Jess tab (O'Connor et al. 2005), a meaning-preserving translation from SWRL rules to Jess rules is possible. In case that a Jess rule produces new OWL individuals, however, Jess does not have any means to "replay" TBox consistency checking or ABox realization, since this would require predicate variables in entailment rules, similar to Hayes (2004) and ter Horst (2005).

OWLIM and Jena, the rule languages from Pellet and Racer, as well as Jess are essentially *forward-chaining* or *data-driven* inference engines, meaning that they start from initial facts (here: RDF triples) and permanently apply the rules until a *fixpoint*

is reached, i.e., until no more information can be added. Forward chaining is a way to carry out all inferences at compile time, even useless inferences for the application. The resulting fixpoint is often called the *deductive closure* and since all information is simultaneously available, it can be queried very effectively at run time. *Backward reasoners*, such as Prolog, clearly do not produce such a large search space, since they are *goal-driven*. However, since inferencing will happen at run time, querying information in a backward reasoner will often be magnitudes slower than in forward engine.

Many forward-chaining engines, such as Jena or Jess are based on the famous RETE algorithm (Forgy 1982). We are not sure whether Ontotext's extremly fast OWLIM framework that will handle most of the workload in our hybrid system is a RETE-based system, but it will probably employ techniques such as those listed above.

Forward chaining as such must not always be a good choice. Known problems are

- potentially large deductive closure
- counting/arithmetics and dynamic creation of structures might lead to non-termination
- cardinality constraints are hard to encode
- checking for the consistency of generated model is achieved by querying for owl:Nothing (could at the same time be a blessing, see later)

However, forward chaining has a number of big advantages and as long as a system scales up well in practice, it is fine to encode reasoning rules as forward-chaining rules:

- basic idea easy to implement
- practically no inference at run time, only indexing
- fast
- terminating in case new individuals are *not* introduced
- storage/access layer: from in-memory, XML-DBs, RDMS, AllegroGraph, ...
- essentially Datalog language ("function-free" Prolog).

On the base of such considerations, MUSING is proposing a reasoning architecture. In the proposed system architecture, input to the system at the moment either comes from natural language text or from XBRL balance sheets, which are the base for ontology population. The initial ontology (essentially the TBox) is checked for consistency by a full description logic reasoner (Pellet: OWL DL). This ontology then is forwarded to the main reasoning component, whose storage model is based on Sesame. ABox equational reasoning is performed via Ontotext's OWLIM, whereas numerical constraints and arithmetics are handled solely by the Jena engine form HP, due to the fact that OWLIM does not have such descriptive means. At the same time, Jena is by a large margin slower than OWLIM. Because OWLIM and Jena perform forward-chaining reasoning, the fixpoint computation stretches over the sequence of the two reasoners in order to reach the deductive closure. Note that since logical variables in both OWLIM and Jena rules only bind one individual at a time, rule knowledge that is based on the existence of all individuals match a logic variable in a specific clause can not be easily formulated within these formalisms. To do so, however, we emulate this behavior in by firstly posting queries to the ontology,

followed secondly by the construction of individuals from the result table generated by the answer to the query, and thirdly by entering these new individuals to the ontology at last (populating). This again might trigger a new fixpoint iteration in OWLIM/Jena.

4 First Conclusions and further work

The paper has presented the actual approach of the MUSING project for integrating semantic resources and tools for the purpose of semantic-based Business Intelligence applications. Re-usability and interoperability are a core concern of our work in this field. MUSING ontologies are continuously updated by domain experts, reacting on the needs generated by specific applications. But we are starting to investigate the use of rule-based and statistical ontology learning methods. Concerning the reasoning platform we briefly presented, our next actions will consist in a thorough evaluation of the platform on the base of a relevant amount of data in the knowledge base generated by the MUSING applications. Till now we used for a first evaluation an external ontology, consisting of approximately 1,000 classes and 20,000 instances. In this context we could already see that the computation of the deductive closure was running much faster with our proposed architecture as with an isolated tool.

5 References

James F. Allen & Alan M. Frisch. What's in a semantic network. Proceedings 20th ACL, 19-27, 1982.

Franz Baader & Ulrike Sattler. An Overview of Tableau Algorithms for Description Logics. Studia Logica, 69, 5–40, 2001.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi & Peter Patel-Schneider. The Description Logic Handbook. Cambridge University Press, 2003.

Mira Balaban. The F-Logic Approach for Description Languages. Annals of Mathematics and Artificial Intelligence, 15(1), 19–60, 1995.

Tim Berners-Lee, James Hendler & Ora Lassila. The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American, May 2001.

Ronald J. Brachman. On the epistemological status of semantic networks. In: Associative Networks: The Representation and Use of Knowledge of Computers, N.V. Findler (ed.). Academic Press, 1979.

Ronald J. Brachman & James G. Schmolze. An overview of the KL-ONE Knowledge Representation System. Cognitive Science 9, 171-216, 1985.

Jeen Broekstra, Arjohn Kampman & Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF. Proceedings International Semantic Web Conference, ISWC 2002.

Dan Brickley & R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/rdf-schema/).

Thierry Declerck & Hans-Ulrich Krieger. Translating XBRL Into Description Logic. An Approach Using Protégé, Sesame & OWL. 455–467, BIS 2006.

Charles Forgy. Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence 19, 17–37, 1982.

Erich Grädel, Phokion G. Kolaitis & Moshe Y. Vardi. On the Decision Problem for Two-Variable First-Order Logic. Bulletin of Symbolic Logic, 3(1), 53--69, 1997.

Volker Haarslev & Ralf Möller. Description of the RACER System and its Applications. Proceedings International Workshop on Description Logics (DL-2001), 131–141, 2001.

Patrick Hayes. RDF Semantics. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/rdf-mt/).

Jerry Hobbs. An OWL Ontology of Time. Draft version, July 2004.

Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof & Mike Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 21 May 2004 (http://www.w3.org/Submission/SWRL/).

Ian Horrocks, Oliver Kutz & Ulrike Sattler. The Even More Irresistible SROIQ. 10th International Conference on Principles of Knowledge Representation and Reasoning, 2006.

Michael Kifer, Georg Lausen & J Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. JACM, 42(4), 741–843, 1995.

Atanas Kiryakov. OWLIM: balancing between scalable repository and light-weight reasoner. Developer's Track, WWW 2006.

Graham Klyne & Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/rdf-concepts/).

Vladimir Kolovski, Bijan Parsia & Evren Sirin. Extending SHOIQ(D) Tableaux with DL-safe Rules: First Results. Proceedings International Workshop on Description Logic, DL-2006.

Hans-Ulrich Krieger. SDL–A Description Language for Building NLP Systems. Proceedings HLT-NAACL Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS), 84–91, 2003.

Hans-Ulrich Krieger. SDL–A Description Language for Specifying NLP Systems. Proceedings 3rd AMAST Workshop on Algebraic Methods in Language Processing, AMiLP-3, 2003.

Frank Manola & Eric Miller. RDF Primer. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/rdf-primer/).

Deborah L. McGuinness & Frank van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/owl-features/).

Martin O'Connor, Holger Knublauch, Samson Tu, Benjamin Grosof, Mike Dean, William Grosso & Mark Musen. Supporting Rule System Interoperability on the Semantic Web with SWRL. Proceedings International Semantic Web Conference (ISWC), 2005.

OWL 1.1. OWL 1.1 Web Ontology Language (http://www.webont.org/owl/1.1/).

Peter F. Patel-Schneider, Patrick Hayes & Ian Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/owl-semantics/).

Eric Prud'hommeaux & Andy Seaborne. SPARQL Query Language for RDF. W3C Candidate Recommendation 14 June 2007 (http://www.w3.org/TR/rdf-sparql-query/).

Racer 2005. RacerPro User's Guide, Version 1.9. December 8, 2005.

Dave Reynolds. Jena Rules tutorial. PowerPoint presentation. Jena User Conference 2006.

Dave Reynolds. Java 2 Inference support. Version 1.35, 2007/03/23, see Web page http://jena.sourceforge.net/inference/

Konstantinos Sagonas, Terrance Swift, David S. Warren, Juliana Freire, Prasad Rao, Baoqiu Cui, Ernie Johnson, Luis de Castro, Rui F. Marques, Steve Dawson & Michael Kifer. The XSB System. Version 3.0. Volume 1: Programmer's Manual. July 26, 2006.

Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. 1st International Conference on Principles of Knowledge Representation and Reasoning, 421–431, 1989.

Theodore Sider. Four Dimensionalism. Philosophical Review 106, 197–231, 1997.

Evren Sirin, Bijan Parsia, Bernardo Cuenca-Grau, Aditya Kalyanpur & Yarden Katz. Peller: A Practical OWL-DL Reasoner. Journal of Web Semantics, 5(2), 2007.

Michael K. Smith, Chris Welty & Deborah L. McGuinness. OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004 (http://www.w3.org/TR/owl-guide/).

SwiftOWLIM 2007. SwiftOWLIM Semantic Repository for RDF(S) and OWL. ver. 2.9.0, 12 June 2007 (http://www.ontotext.com/owlim/OWLIMSysDoc.pdf)

Herman J. ter Horst. Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. Proceedings ISWC, 668-684, 2005.

Ivan Terziev, Atanas Kiryakov & Dimitar Manov. D1.8.1 Base upper-level ontology (BULO) Guidance1, EU-IST Project IST-2003-506826 SEKT, WP1, D1.8.1, 2005, (http://proton.semanticweb.org/D1_8_1.pdf)

Christopher Welty, Richard Fikes & Selene Makarios. A Reusable Ontology for Fluents in OWL. IBM Research Report, RC23755 (W0510-142), October 21, 2005.

William A. Woods. What's in a link. Foundations for semantic networks. In: Representation and Understanding, D.G. Bobrow & A.M. Collins (eds.). Academic Press, 1975.

Guizhen Yang, Michael Kifer, Chang Zhao & Vishal Chowdhary. Flora-2: User's Manual, Version 0.94 (Narumigata). April 30, 2005.

Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195--197 (1981)