Reuse of thesauri by means of Semantic Web technology

by

Bert De Winter
(Studentno. 836645461)

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science (Ir.) in Technical Informatics
Samenvatting

Abstract

1 Problem definition

1.1 Organizational context

1.1.1 Agfa

1.1.2 Mission of RDM

1.2 Statement of the problem

2 Assignment

2.1 Introduction

2.2 Research questions

2.3 Boundary conditions

2.4 Build-up of this report

2.5 Research method

2.6 Motivation and benefits of the research

3 Research context

3.1 The semantic web

3.1.1 Vision

3.1.2 Architecture

3.1.2.1 URI

3.1.2.2 XML

3.1.2.3 RDF and RDF Schema

3.1.2.4 Ontology

3.1.2.4.1 Overview of OWL

3.1.2.4.2 The role of ontology’s for the World Wide Web

3.1.2.5 Logic and proof

3.2 Thesaurus and Ontology

3.2.1 Thesauri

3.2.1.1 Thesaurus: history and use

3.2.1.2 Standards

3.2.1.2.1 ISO 2788

3.2.1.2.1.1 Scope and field of application

3.2.1.2.1.2 Concepts

3.2.1.2.1.3 Relationships

3.2.1.2.2 ISO 5964:1985

3.2.1.2.3 ANSI/NISO Z39.19:2003

3.2.1.3 Thesaurus and ontology: comparison

3.2.1.4 The Ontology spectrum

3.2.1.5 Migration from thesauri to ontology’s

3.3 Semantics

3.3.1 Mind-brain problem

3.3.2 Description and specification of semantics

3.3.3 First order predicate logic

3.3.3.1 Language constructions

3.3.3.2 Predicate logic: semantics

3.3.4 Semantics and the Semantics web

3.3.4.1 Real world semantics

3.3.4.2 Model theoretic semantics

3.3.4.2.1 Model theory
7 Modeling thesauri using Semantic Web languages ........................................... 61

7.1 Introduction ................................................................................................. 61
7.2 Term-based thesaurus model ............................................................... 61
7.3 Concept-based thesaurus model .......................................................... 63
7.4 OWL and thesauri ..................................................................................... 64

7.4.1 OWL features ......................................................................................... 65

7.4.1.1 OWL Light features ............................................................................. 65
7.4.1.1.1 Cardinality ......................................................................................... 65
7.4.1.1.2 Equality - Inequality ......................................................................... 65
7.4.1.1.2.1 Equivalent class ............................................................................ 65
7.4.1.1.2.2 SameAs ......................................................................................... 65
7.4.1.1.3 Property characteristics .............................................................. 66
7.4.1.1.3.1 InverseOf ....................................................................................... 66
7.4.1.1.3.2 TransitiveProperty ....................................................................... 66
7.4.1.1.3.3 SymmetricProperty ..................................................................... 66
7.4.1.1.3.4 FunctionalProperty .................................................................... 66
7.4.1.1.3.5 InverseFunctionalProperty ....................................................... 66
7.4.1.1.4 Annotations ...................................................................................... 66
7.4.1.1.5 Versioning ......................................................................................... 67
7.4.1.2 OWL DL features .............................................................................. 67
7.4.1.2.1 disjointWith ..................................................................................... 67
7.4.1.2.2 unionOf, complementOf, intersectionOf ....................................... 67

7.5 Proposal for a generic OWL thesaurus model ........................................ 68

7.5.1 Classes ..................................................................................................... 68
7.5.1.1 Object Properties ............................................................................... 69
7.5.1.1.1 Associative relationships ............................................................. 69
7.5.1.1.2 Equivalence relation ..................................................................... 70
7.5.1.1.3 Hierarchical relations .................................................................. 70
7.5.1.2 Data type properties ......................................................................... 71
7.5.2 Non standard properties ...................................................................... 71
7.5.2.1 INVENT.DE ....................................................................................... 72
7.5.2.2 TELEF,REGIST ................................................................................... 72
7.5.2.3 SECURITY .......................................................................................... 73

7.6 Discussion .................................................................................................... 73

8 Thesaurus services .......................................................................................... 75

8.1 Introduction .................................................................................................. 75
8.2 Generic services ........................................................................................... 75
8.3 Reasoning ..................................................................................................... 75
8.3.1 OWL reasoners ...................................................................................... 76
8.3.1.1 Racer .................................................................................................. 76
8.3.1.2 Euler ................................................................................................... 76
8.4 RDF/RDFS querying .................................................................................. 76
8.4.1 Sesame .................................................................................................... 77
8.4.2 CWM ...................................................................................................... 77
8.5 Discussion and Conclusion ....................................................................... 78

9 Test of the RDF(S)/OWL thesaurus ................................................................ 80
9.1 Introduction........................................................................................................ 80
9.2 Test set-up........................................................................................................ 80
9.3 Test with Racer................................................................................................. 81
9.4 Test with Euler................................................................................................. 81
9.5 Test with Sesame............................................................................................. 81
9.6 Tests with CWM............................................................................................... 82
9.7 Discussion and conclusion............................................................................... 82
10 Conclusions and future work........................................................................... 84
11 Appendix........................................................................................................... 87

11.1 Python source code for conversion of the thesaurus from Basis text format to a simple XML representation......................................................... 87
11.2 The GEM RDF Schema thesaurus model....................................................... 89
11.3 The CERES/NBII Thesaurus RDF Schema................................................... 91
11.4 RDF/XML Thesaurus Schema......................................................................... 93
11.5 RDF/XML thesaurus model for Sesame......................................................... 97
11.6 Testset1........................................................................................................... 100
11.7 Racer Test results.......................................................................................... 103
  11.7.1 Test 1........................................................................................................ 103
  11.7.2 Test 2........................................................................................................ 104
  11.7.3 Test 3........................................................................................................ 104
  11.7.4 Test 4........................................................................................................ 104
  11.7.5 Test 5........................................................................................................ 104
  11.7.6 Test 6........................................................................................................ 106
  11.7.7 Test 7........................................................................................................ 106
  11.7.8 Test 8........................................................................................................ 106
  11.7.9 Test 9........................................................................................................ 107
  11.7.10 Test 10................................................................................................. 108
  11.7.11 Test 11................................................................................................. 109
11.8 Euler Test results............................................................................................ 109
  11.8.1 Test 1........................................................................................................ 109
  11.8.2 Test 2........................................................................................................ 110
  11.8.3 Test 3........................................................................................................ 110
  11.8.4 Test 4........................................................................................................ 111
  11.8.5 Test 5........................................................................................................ 111
  11.8.6 Test 6........................................................................................................ 113
  11.8.7 Test 7........................................................................................................ 114
  11.8.8 Test 8........................................................................................................ 114
  11.8.9 Test 9........................................................................................................ 116
  11.8.10 Test 10................................................................................................. 116
  11.8.11 Test 11................................................................................................. 118
11.9 CWM Test results......................................................................................... 118
  11.9.1 Test 1........................................................................................................ 118
  11.9.2 Test 2........................................................................................................ 119
  11.9.3 Test 3........................................................................................................ 120
  11.9.4 Test 4........................................................................................................ 120
  11.9.5 Test 5........................................................................................................ 121
  11.9.6 Test 6........................................................................................................ 122
  11.9.7 Test 7........................................................................................................ 123
  11.9.8 Test 8........................................................................................................ 124
Acknowledgements
I would like to express my gratitude to all those who gave me the possibility to complete this master thesis.

I want to thank the management of Agfa’s Research and Development Materials (RDM) department and RDM/CS for giving me permission to commence this thesis in the first instance, to do the necessary research work and to use departmental data.

My colleagues from RDM/CS and GICS/ICAS/RDM supported me in my research work. I want to thank them for all their help, interest and valuable hints. Especially I am obliged to Albers S., Beets L., De Beuckelaer D; Schaerlaecken M., Van Hertbruggen B. and Broeckx M. for there advise, help and technical support.

I am deeply indebted to my supervisors De Roo J. from Agfa’s Healthcare division, Dr. Van Veen M. and Prof. Dr. S. Joosten from the ‘Open Universiteit Nederland’ whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.

I would also like to thank the people from the ‘Vrije Universiteit Amsterdam’ who helped me a lot in becoming familiar with Semantic Web based knowledge representation and the review of this work, more especially Prof. Dr. Van Harmelen, Dr. Stuckenschmidt H., and van Assem M.

Especially, I would like to give my special thanks to my wife Marleen and my daughters Karolien and Liesbeth whose patience enabled me to complete this work.
Samenvatting

In dit werk werd onderzocht hoe op de ISO standaarden gebaseerde thesauri, kunnen herbruikt worden met behulp van technologie die ontwikkeld wordt in kader van het Semantische Web. Het doel van het onderzoek was om enerzijds na te gaan wat een goede conversie methode is, en anderzijds of het mogelijk is een thesaurus meta model te definiëren met behulp van RDF(S)/OWL zodat algemene redeneerssoftware de gewenste thesaurus diensten kan aanbieden. De praktische bruikbaarheid van het systeem stond daarbij centraal aangezien hierover weinig is terug te vinden in de literatuur. Het betreft hier dan ook een experimenteel onderzoek waarbij een thesaurus van een 50.000 termen werd gevonverteerd en de thesaurus diensten geïmplementeerd werden met behulp van twee RDF(S) reasoners en twee OWL reasoners.

The migratiemethode die in dit werk wordt voorgesteld, is gebaseerd op eerder onderzoek maar werd verder verfijnd. Het uitgangspunt van deze methode is het bestaan van een gestandaardiseerd RDF(S)/OWL thesaurus meta-model. Hiervoor werd een term gebaseerd OWL/RDF(S) thesaurus model voorgesteld. Dit in tegenstelling tot het SKOS kern meta model van W3C dat conceptgebaseerd is. De reden hiervoor is dat een concept gebaseerde aanpak geen duidelijke voordelen heeft en problemen kan introduceren bij het converteren van bestaande thesauri. De voorgestelde methode bestaat uit drie stappen. In de eerste stap wordt de bestaande thesaurus geanalyseerd om zo het thesaurus metamodel duidelijk te kunnen beschrijven en af te beelden op het meta model van het standaard RDF(S)/OWL model waarbij de betekenis van het oorspronkelijke metamodel overandert moet blijven. Deze stap wordt daarom de semantische afbeelding genoemd. Tijdens deze activiteit wordt het ook duidelijk of het generieke RDF(S)/OWL model nog verder dient uitgebreid te worden om zo ook thesaurusspecifieke relaties tussen termen te kunnen representeren.

De volgende twee stappen zijn zuiver syntactische vertalingen van de thesaurus data naar de nieuwe datastructuren waarbij gebruik wordt gemaakt van de terminologie van het (eventueel aangepaste) standaard RDF(S)/OWL thesaurus meta model. Hierbij werden XML hulpmiddelen gebruikt om de conversie eenvoudig en aanpasbaar te houden. Verder werden algemene thesaurus diensten beschreven en geïmplementeerd met behulp van verschillende generieke OWL en RDF(S) redeneerssoftware.

The volledige methode werd uitgetest door conversie van een thesaurus van 50.000 termen. De uitgesteste OWL redeneerssoftware Racer en Euler was niet in staat om de gewenste thesaurus diensten te leveren voor de complete thesaurus van 50.000 termen. Waarschijnlijk was de rekencomplexiteit te hoog. Verder onderzoek is nodig om dit duidelijk te maken.
De geteste RDF(S) redeneerssoftware, CWM en Sesame konden dit wel. Een nadeel is wel dat met RDF(S) niet alle applicatiesemantiek kan vastgelegd worden zodat deze moet hard gecodeerd worden in de applicatie die gebruik maakt van de RDF(S) redeneerssoftware.
Abstract

The purpose of this work was to investigate how ISO standard based thesauri can be reused by means of semantic web technology. Besides the proposal of a conversion method, the goal was also to investigate if it is possible to define a thesaurus meta model using RDF(S)/OWL in a way that generic reasoners are able to deliver the desired thesaurus services, for thesauri of realistic size as there is not much information available about this topic in literature. Focus was on the practical usefulness of the resulting thesaurus system and therefore a thesaurus of 15000 terms has been converted. The thesaurus services were tested using two OWL reasoners and two RDF reasoners.

The migration method proposed in this work is based on previous work of but has been streamlined. The point of departure of the in this work proposed method is the existence of a generic RDF(S)/OWL thesaurus meta-model. A so called term based OWL/RDF(S) thesaurus model has been proposed. This in contrast with the SKOS core meta model of W3C. The rationale for this is that the concept based approach has no clear benefits and even can introduce some new problems converting an existing thesaurus.

The proposed method is a simple three step process to convert the existing thesaurus data to RDF assertions with model theoretic semantics defined by the RDF(S)/OWL meta model.

The first step of the method is an analysis of the existing thesaurus to indentify the source thesaurus meta-model semantics to be able to define a mapping between the source meta model and the destination meta model in a semantics preserving way. This is called a semantic mapping. During this mapping activity it becomes clear if the generic OWL/RDF(S) meta model needs to be extended in order to be able to represent also specific relations between terms. The two following steps are then pure syntactic conversions of the thesaurus data to the new data structure using the terminology of the RDF(S)/OWL thesaurus meta model and XML tools to simply the conversion task.

Generic thesaurus services have been described and implemented by means of different generic OWL and RDF(S) reasoners. The complete method was proved by converting and testing a thesaurus of 15000 terms. The tested OWL reasoners Racer and Euler were not capable to cope with this thesaurus. Probably, the computation complexity was too high. Further research is needed to pin-point the exact causes and to indicate some possible solutions.

But both tested RDF(S) reasoners, CWM and Sesame, could deliver the desired thesaurus services also when a complete thesaurus of about 150,000 terms was loaded. A consequence of this approach is that some ‘knowledge’ of the thesaurus model which can only be described with OWL, must be hard coded in the application interfacing to the RDF(S) reasoner.
1 Problem definition

1.1 Organizational context

1.1.1 Agfa
The Agfa-Gevaert Group operates with photographic and digital systems in the fields of information, communication, health and safety, leisure and recreation. Agfa’s operational activities are classified in 3 business segments, which are further divided into 5 business groups:

- Graphic Systems: digital and analogue pre-press systems, software and consumables.
- Technical Imaging:
- HealthCare: medical imaging and information management systems for healthcare.
- Non-Destructive Testing: X-ray and ultrasonic systems for safety and quality controls of all materials without damaging or deforming them.
- Industrial Imaging: micrographics and document systems, motion picture film and high-security identification documents.
- Consumer Imaging: photographic products for the consumer market and equipment and consumables for finishing labs.

Agfa has also some global shared service departments. One of them is Research and Development Materials (RDM).

1.1.2 Mission of RDM
RDM strives to provide competitive advantage to Agfa Business Units. Predominantly chemical knowledge as well as hard- and software capabilities will enable RDM to:
- Develop new and improved imaging products for existing market segments.
- Develop new products for new market segments.

RDM strives to develop these products on time and within budget in order to:
- Be regarded as a leading, reliable partner for conceiving and developing new products.
- Make new products an Agfa core activity.

1.2 Statement of the problem
Knowledge sharing and communication are of paramount importance in any organization, certainly in a R&D environment.

To support product development activities, RDM has build out over the last 20 years a knowledge base with a lot of valuable information:
management and cost information of RDM projects, technical and scientific library catalogue, administrative reports, technical reports, regulations, synthesis tree, protocols, market surveys, safety reports, chemical products and properties. 

This knowledge base is built on the BASIS DBMS from OpenText. BASIS offers a thesaurus with support for all thesaurus relationships defined in ANSI Z39.19. 

The thesaurus is used for classification of the information and browsing. An important feature is the possibility to expand search terms during information retrieval to hyponyms and synonyms. Another big benefit is the support for assigning metadata to documents. A drawback of the thesaurus is that the supported relationships are limited and generic and often used with redundant semantics. 

Although a lot of valuable information is stored in the knowledge system, a huge number of documents is nowadays residing outside the BASIS system and this number is increasing rapidly: 

- documents on the file systems 
- recipes in the Lassy en Kobra formulation information systems 
- analysis reports in the IMPALA - Laboratory Information Management system 
- documents on the intranet 
- documents and information in the upcoming Electronic Laboratory Notebooks (ELN), e.g. Project Manager Tool (PMT) 
- documents in QuickPlaces 
- documents in Lotus Notes databases 

It is clear that the complexity of the information architecture has significantly increased over the last decade. Finding back the desired information is time-consuming. Knowledge of the existing information architecture is a prerequisite to successfully find back the desired information. Also knowledge of the specific vocabulary used in the different information systems is important to find back all information. E.g. the same chemical product has often different names in the various information systems. 

At the moment, the RDM intranet provides a ubiquitous interface to RDM’s information and knowledge. Existing keyword-based searches can retrieve irrelevant information when different terms with the same meaning about the desired content are used. 

There are still other serious problems associated with the use of the intranet search engine: 

- low recall: important and relevant pages are not retrieved 
- low precision: even if the most important pages are retrieved together with a lot of irrelevant pages, this is still like searching for a needle in a haystack.
- search results are highly sensitive to vocabulary. Keyword-based searches can retrieve irrelevant information when different terms with the same meaning about the desired content are used.

- Automatic content integration is not possible at the moment. New documents are often created with information from different information systems. E.g. an experiment report could contain recipes out of Lab Support System combined with chemical structures out of Apache and analysis results coming from the LIMS system. Automatic generation of these documents would save a lot of time and hides the complex information architecture for the researchers.

- No classification of information possible. There is no classification of the documents possible at the moment, while this would be very helpful for humans browsing for information. Having a kind of taxonomy supported browsing and search covering all the information sources would be a big advantage and very helpful for information retrieval and exchange.

The same problems are existing on the World Wide Web were the enormous amount of data has made it increasingly difficult to find, access, present and maintain relevant information. In response to this problems, many new research initiatives have been set up. Tim Berners-Lee, Director of the World Wide Web Consortium, referred to the future of the current Web as the Semantic Web, which will be further explained in chapter 3.
2 Assignment

2.1 Introduction

Thesauri are widely used as an aid to information retrieval, also at Agfa. It are highly valuable knowledge resources. Reuse of thesauri within the web environment could for instance enhance web search. OWL-based representations for thesaurus data could be the enabling step towards this goal. Some proposals for RDF schema’s to represent specific thesauri have been made, see also section 4.1. However, many issues remain to be considered. For example, although there is standardization in this area (see section 3.2.1.2), in practice thesauri come in different flavors. A useful schema must be able to accommodate these variations in meaning.

Thesauri could be reused by means of semantic web technology in a way that the meta-model semantics of the existing thesaurus meta-model is not only preserved, but also explicitly described using OWL constructs which have precise model theoretic semantics as explained in section 3.3. This way they can provide basically the same services as the original thesauri using a generic reasoner and without the need for hard coding the thesaurus meta-model semantics into the thesaurus application.

But thesauri have limitations stemming from the relationships as defined in the ISO standards having a very general meaning. This leads often to the ambiguous use of the relationships in a thesaurus: e.g. ‘related term’ is used to connect terms when no other relationships could be used to do this. They lack possibility to represent ‘more’ meaning explicitly and formally represented in a way this description could be used by machines. The aim of the semantic web is just to do this. An OWL thesaurus schema could be defined with clear model theoretic semantics for the representation of not only the ISO standard thesaurus relations but also part of the semantics introduced by the implementation of a thesaurus as intended by the thesaurus builder. This way a thesaurus could be transformed into an ontology.

Just having a valid and flexible OWL schema for the representation of thesauri is not solving the complete conversion problems. As reported by conversion of existing thesauri could introduce unwanted semantics inherited form the use of RDF(S) or OWL features and must be carried out with care. Another question is what is needed further to construct a practical useful OWL thesaurus delivering the desired thesaurus services based on semantic web technology.
2.2 Research questions

The considerations formulated in section 2.1 leads to the following central question of this work:

*How can ISO standard based thesauri be reused by means of semantic web technology?*

This question leads to the following sub-questions:

1. How can we elicit the structure and the semantic model of existing thesauri?
2. What is a minimal set of thesaurus services a thesaurus system should be able to deliver?
3. What is a useful RDF(S)/OWL schema to represent thesauri, preserving the meta-model semantics of a standard thesaurus, while it still is possible to add additional features with other semantics.
4. Which steps are needed to convert an existing thesaurus to the new data structure.
5. Are generic reasoners able to deliver the desired thesaurus services, also under practical conditions.

2.3 Boundary conditions

The most important boundary conditions related to a possible solution are:

- The conversion method must accurately preserve the meta-model semantic of the source thesaurus.
- Although the focus is on ISO standards based thesauri, the OWL model must accommodate variations in thesaurus structure, semantics and usage because this occurs frequently in implemented thesauri.
- The RDF(S)/OWL ontology model should support extension of its relationship set in order to have the possibility to enrich the semantics of the meta-model and to evolve towards an ontology.
- The RDF(S)/OWL meta-model should represent the minimal model theoretic semantics in order to permit a generic reasoner to deliver the desired thesaurus services.
- The reasoner should deliver the desired thesaurus services within a limited amount of time so that the system is useful under practical conditions.

2.4 Build-up of this report

This report contains three main parts. The first part, which encloses chapters 1 and 2, where also this section is part of, is used to describe the problem and to sketch the research assignment.
In the second part, which encloses chapter 3 and 4, the results of a literature study will be reported. The goal of this chapters is to paint a picture of the research context and to clarify the scientific theory which will be needed for the actual research. But also to focus on work related to this research: Chapter 4 describes semantic web representation formalisms for thesauri and methods for the conversion of existing thesauri to web ontology’s.

The third section, enclosing chapters 5, 6, 7, 8 and 8.5 is reporting about the actual research work. First the method used to convert the existing thesauri to an OWL representation is explained in chapter six which will give an answer to sub question 4 as formulated in section 2.2. Chapter 7 contains an analysis of the Agfa thesauri which gives answers to sub question 1. In chapter 7, an extensible OWL thesaurus model is proposed giving answer to sub question 3, while in chapter 8 generic thesaurus services are described (sub question 2). Chapter 8.5 reports about the implementation of the semantic web based thesaurus and the tests with various reasoners implementing the defined thesaurus services, which is focusing mainly on sub question 5.

2.5 Research method
An experimental research design has been chosen. As stated in section 2.2 it is important to have a system which is practical useful. This will be achieved by setting up tests to prove that the system based on the OWL/RDF(S) thesaurus is able to deliver the desired thesaurus services. Additional benefits of this test driven experimental approach is that this way the assumptions made during the design of the conversion method and thesaurus meta-model can be tested for validity. The behavior of the system in real-life conditions can be investigated, which is important towards a practical application of the method. Within this experimental approach also different components will be tested e.g. query engines because at the moment there is not much literature available about the practical realization of a OWL based thesaurus system.

A complete theoretical approach would not permit to draw conclusions about the operational behavior of the OWL/RDF(S) reasoners. Also a complete design approach is at the moment to early. Emphasis in this work is on a good semantic web based representation model for thesauri and a valid method to convert existing resources with a good indication of practical usability and less on having a complete and working system.

2.6 Motivation and benefits of the research
Thesauri could be reused by means of semantic web technology by preserving the existing thesaurus meta-model semantics. This way, they can provide basically the same services as the original thesauri. This could deliver to the following benefits:
It fits in a strategy to use intranet as an ubiquitous interface to the information and knowledge of a company.

Use of open standards has strategic benefits, there is a high probability that a lot of tools (open source and commercial) for semantic web-based knowledge management will become available in the future. Such tools provide facilities for finding, sharing, summarizing, visualizing, browsing and organizing knowledge and could then be used.

Replacement possible of legacy system with standard based system

Compatible with web technology

More easy (web standard based) interfacing or integration of thesaurus 'knowledge' to other systems possible

Using OWL, with a well defined model theoretic semantics, has following expected benefits:

- The OWL thesaurus meta-model could be made with explicit representation of not only the ISO standard semantics but also the semantics introduced by the thesaurus implementation and part of the semantics as intended by the thesaurus builder.
- Generic OWL reasoners could be used to deliver the desired thesaurus services but also the specific services connected to the non standard features of the thesaurus because they can interpret the semantics of the OWL thesaurus.
- Generic OWL reasoners could be used to detect internal inconsistencies in the thesaurus.
- By using OWL with its foundations in Description Logics, not all knowledge must be explicitly available in the thesaurus model, a reasoner can deduce new facts based on incomplete information.
- OWL reasoners could automatically calculate term taxonomies based on the term assertions.
- Based on previous benefits, it can be expected that the management of an OWL thesaurus could be easier to do in comparison with thesauri with no explicitly represented meta-model semantics.
- The OWL thesaurus meta-model could be extended to represent other more specialised relationships. This way, a thesaurus could evolve towards a more expressive ontology.

As a result, the ontology could be a cornerstone in a new semantic web based knowledge management architecture for RDM and Agfa.

The experimental research to find out if it is feasible to have a practical useful semantic web technology based thesaurus system, using OWL to describe the thesaurus meta model and data, and a generic reasoner to deliver the desired thesaurus services, is also motivated by the fact that at the moment not much literature information is available about this topic. Most proposals make use of RDF(S), no test results about the use of different OWL reasoners within this field have been found. See also
chapter 4 were an overview is given about the related work within the research context.
3 Research context

In this chapter the broad research context of this work will be sketched. Section 3.1 will start with the vision and architecture of the Semantic Web. In section 3.2 thesauri will be clarified and compared with ontology’s. Section 3.3 will go more deeply into the meaning of the word semantic, as this word conveys often different meanings also in the context of the Semantic Web. In section 3.4 Description Logics will be expounded because they are important knowledge representation formalisms. They form also the foundation of the Semantic Web ontology languages (OWL).

3.1 The semantic web

The World Wide Web enabled the easy access for man to huge amounts of electronically accessible information. According to the Web nowadays contains around 3 billion static documents. But at the same time, this enormous amount of data has it made increasingly difficulty to find, access, present and maintain relevant information. This state is also known as information overload, a state in which the searcher no longer is able to process the volume of information retrieved effectively. Software agents could assist people in carrying out information retrieval or even more complex tasks, as explained in Scientific American. Unfortunately the data formats (mainly Hypertext Markup Language, HTML) and technology used for the World Wide Web today is not aimed for machine processing. The Semantic Web is proposed as a solution to this problem.

3.1.1 Vision

The semantic web is a further extension of the current World Wide Web (WWW), in which information is given well-defined meaning, better enabling computers and people to work in cooperation. According to , the Semantic Web will provide an infrastructure that enables not just web pages, but databases, services, programs, sensors, personal devices, and even household appliances to both consume and produce data on the web. Software agents can use this information to search, filter and prepare information in new and exciting ways to assist the web user. New languages, making significantly more of the information on the web machine-readable, power this vision and will enable the development of a new generation of technologies and toolkits.

3.1.2 Architecture

The architecture of the semantic web is represented by Figure 1 (Tim Berners-Lee 2002).
3.1.2.1 URI
Uniform Resource Identifiers (URI) provide a simple and extensible means for identifying a resource. Resources can be anything: documents, images, downloadable files, services, electronic mailboxes, and other resources but also abstract concepts. They make resources available under a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail addressable in the same simple way. They reduce the tedium of "log in to this server, then issue this magic command ..." down to a single click.

3.1.2.2 XML
The base layers of the Semantic Web consists of the Extensible Markup Language (XML), encoded in Unicode, an international encoding standard supported by most operating systems and web browsers. XML is a `meta language' —a language for describing other languages—which enables the design of new customized markup languages for limitless different types of documents. XML allows the flexible development of user-defined document types. It provides a robust, non-proprietary, persistent, and verifiable file format for the storage and transmission of text and data both on and off the Web an between applications. XML allows Web-page authors to use there own set of markup tags. According to (Frank van Harmelen and Dieter Fensel 1999)allows XML to structure Web pages as labeled trees, where labels can be chosen by the information provider to reflect as much as possible of the document semantics as is required.
3.1.2.3 RDF and RDF Schema

For machine processing purposes, XML is not sufficient. To make clear that the author of a document is John Doe, one could add the `<author>` tag to John Doe (`<author>John Doe</author>`) but for a software agent it is not clear what the scope is of this information. There is a need for having a language to represent meta data about resources. That is exactly the purpose of the Resource Description Framework (RDF). The broad goal of RDF is to define a mechanism for describing resources in a way that this information is also machine processable. RDF is based on the idea that the things being described have properties which have values, and that resources can be described by making statements. An example of such a statement is "John Doe is author of document X". A concrete syntax is also needed for the purposes of creating and exchanging this meta data. The W3C specification of RDF uses XML to represent RDF, named RDF/XML.

RDF provides a way to express simple statements about resources, using named properties and values. However, RDF user communities also need the ability to define common vocabularies (terms) and this is where RDF Schema can be used.

RDF Schema provides basic capabilities for describing RDF vocabularies.

3.1.2.4 Ontology

Other richer schema capabilities that have been identified as useful (but that are not provided by RDF Schema) include:

- **cardinality constraints** on properties, e.g., that a Person has exactly one biological father.
- specifying that a given property (such as `ex:hasAncestor`) is **transitive**, e.g., that if A `ex:hasAncestor` B, and B `ex:hasAncestor` C, then A `ex:hasAncestor` C.
- specifying that a given property is a unique identifier (or key) for instances of a particular class.
- specifying that two different classes (having different URI refs) actually represent the same class.
- specifying that two different instances (having different URI refs) actually represent the same individual.
- specifying constraints on the range or cardinality of a property that depend on the class of resource to which a property is applied, e.g., being able to say that for a soccer team the `ex:hasPlayers` property has 11 values, while for a basketball team the same property should have only 5 values.
- the ability to describe new classes in terms of combinations (e.g., unions and intersections) of other classes, or to say that two classes are disjoint (i.e., that no resource is an instance of both classes).

The additional capabilities mentioned above, in addition to others, are the targets of **ontology** languages such as DAML+OIL and OWL. Both these languages are based on RDF and RDF Schema and are therefore
represented as a layer on top of the RDF layer in Figure 1. The intent of such languages is to provide additional machine-processable semantics for resources, that is, to make the machine representations of resources more closely resemble their intended real world counterparts.

3.1.2.4.1 Overview of OWL

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

**OWL Lite** supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives.

**OWL DL** supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is so named due to its correspondence with description logics, a field of research that has studied the logics that form the formal foundation of OWL. For a short introduction in DL, see also section 3.4.

**OWL Full** is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

OWL Full can be viewed as an extension of RDF, while OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF. An Important difference is that OWL Full allows the use of classes as instances and OWL DL and Lite does not. Every OWL (Lite, DL, Full) document is an RDF document, and every RDF document is an OWL Full document, but only some RDF documents will be a legal OWL Lite or OWL DL document. Because of this, some care has to be taken when a user wants to migrate an RDF document to OWL. When the expressiveness of OWL DL or OWL Lite is deemed appropriate, some precautions have to be taken to ensure that the original RDF document
complies with the additional constraints imposed by OWL DL and OWL Lite.

3.1.2.4.2 The role of ontology’s for the World Wide Web

According to, the name Ontology has been taken from Philosophy, where it means a systematic explanation of Existence. Later, it is also used in the field of Artificial Intelligence, defined by in 1991 byas follows: "An Ontology defines the basic terms and relations comprising the vocabulary of a topic. Ontology’s aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups."

Ontology’s could be considered as a specification mechanism, a formal representation of domain knowledge.

The role ontology’s can play in the semantic web vision is formulated by W3C (Jeff Heflin 2003) as follows: "Ontology’s figure prominently in the emerging Semantic Web as a way of representing the semantics of documents and enabling the semantics to be used by web applications and intelligent agents. Ontology’s can prove very useful for a community as a way of structuring and defining the meaning of the metadata terms that are currently being collected and standardized. Using ontology’s, tomorrow's applications can be "intelligent," in the sense that they can more accurately work at the human conceptual level."

3.1.2.5 Logic and proof

Inference rules in ontology’s supply further power. With the aid of logic, it is then possible to deduce new facts based on the ontology knowledge and inference rules. How this is achieved is described in more detail in section 3.4 about Description Logics, the fundaments of the web ontology and logic layers. The proof layer provides further the possibility to make the inference process clear to the human users but are out of the scope of this work.

3.2 Thesaurus and Ontology

Ontology’s are a key enabling technology for the Semantic Web according to . They interweave human understanding of symbols with their machine process ability. The reason ontology’s are becoming popular is largely due to what they promise: a shared and common understanding of a domain that can be communicated between people and application systems. Thesauri are related to ontology’s. According to the (2003b) Guidelines for the Construction, Format, and Management of Monolingual Thesauri, a thesaurus is a controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationships.
3.2.1 Thesauri
The structure of thesauri are controlled by international standards, ISO 2788 and ISO 5964. The RDM thesauri are constructed in compliance with the ISO 2788:1986 standard. The data model and semantics of thesauri as defined in ISO 2788 and ISO 5964 will be reviewed.

3.2.1.1 Thesaurus: history and use
The term 'thesaurus' has already a long history. In his book 'Thesaurustechnologie", (Dirk Vervenne 2002) gives a short overview of the history of thesauri.
According to Websters dictionary (ref) a thesaurus is "a book containing a store of words or information about a particular field or set of concepts".
From an historical point of view, the idea of a thesaurus as a list of terms is already very old. There are examples of systematic lists of terms, which are over 5000 years old. More recently in 1911, P.M. Roget realized a thesaurus for the classification of synonyms of the English language.
Librarians started using thesauri for the purpose of information retrieval already from the beginning of the twentieth century. The first international standard for the establishment and development of monolingual thesauri was published in 1974 (1986b).

lists the main uses of thesauri in Information Retrieval:
1. Thesauri provide a map of a given field of knowledge, showing concepts and relations.
2. Thesauri provide a standard vocabulary for consistent indexing.
3. Thesauri assist users with locating terms for proper query formulation. (An example of a query would be the words chosen as input to a web search engine).
4. Thesauri help ensure that only one term from a synonym set is used for indexing and searching: otherwise a searcher who uses one synonym and retrieves some useful documents may think the correct term has been used and the search has been exhaustive, without knowing that there are other useful documents under other synonyms.
5. Thesauri provide classified hierarchies for broadening or narrowing a search (selecting query terms which are broader or narrower in meaning) if too many or too few documents are retrieved at the first attempt.
Thesauri have been used for decades for documentation, categorization and retrieval in library archives and databases.

3.2.1.2 Standards
The ISO standards for thesauri (ISO 2788 and ISO 5964) and NISO standard Z39.19:2003 will be reviewed.

3.2.1.2.1 ISO 2788
ISO 2788 recommends documentation guidelines for the establishment and development of monolingual thesauri.
3.2.1.2.1.1  Scope and field of application

The recommendations set out in this International Standard are intended to ensure consistent practice within a single indexing agency, or between different agencies (for example members of a network). They should not be regarded, however, as mandatory instructions. As far as possible the techniques described in this International Standard are not limited to a particular method of indexing, whether post-coordinate or pre-coordinate. This International Standard is, however, subject to the following restrictions:

- it deals with the display and organization of terms that form a controlled subset of natural language. It does not suggest procedures for organizing and displaying mathematical and chemical formulae.
- it is generally based on the concept of "preferred terms".
- its application is limited to agencies in which human indexers are used to analyze documents and express their subjects in the terms of a controlled indexing language. It is not applicable to these agencies which apply entirely automatic indexing techniques.
- it deals mainly with procedures for indexing collections of documents listed in catalogues or bibliographies.

The recommendations contained in this International Standard are related to monolingual thesauri, without reference to the special requirements of multilingual thesauri, i.e. those thesauri in which conceptual equivalences are expressed in terms selected from more than one natural language. Two principal means for achieving vocabulary control are employed in thesauri:

1. terms are deliberately restricted in scope to selected meanings. By using hierarchical relationships, the intended meaning of a term is indicated. This way homonyms can be distinguished. Scope notes (SN) are further appended to terms if this is not sufficient.
2. preferred terms are used consistently for indexing when a concept can be expressed by two or more terms.

3.2.1.2.1.2  Concepts

In a thesaurus, concepts are represented by indexing terms, preferably in the form of a noun or noun phrase. A preferred term is a term used consistently when indexing to represent a given concept. A non preferred term is a synonym or quasi synonym of a preferred term, provided as an entry point in a thesaurus, directing the user by an instruction to the preferred term.

The standard classifies the concepts into categories:

Table 1: thesaurus concepts

<table>
<thead>
<tr>
<th>Concrete entities</th>
<th>Things (e.g. chair)</th>
</tr>
</thead>
</table>
3.2.1.2.1.3 Relationships

Relationships of two different kinds can occur in a thesaurus: relationships between individual concepts and inter-category relationships. Inter-category relationships are imposing an overall structure or macro classification to ensure that similar concepts are brought together.

Three classes of inter-term relationships are recognized in thesauri: The equivalence relationship, the hierarchical relationship and the associative relationship. The thesaurus relationships are summarized in table 2.

**Table 2: relationships**

<table>
<thead>
<tr>
<th>Class</th>
<th>abrev.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>equivalence</td>
<td>USE</td>
<td>Use, Prefix to the preferred term</td>
</tr>
<tr>
<td></td>
<td>UF</td>
<td>Used for, Prefix to the non-preferred term</td>
</tr>
<tr>
<td>hierarchical</td>
<td>TT</td>
<td>Top term in a hierarchy</td>
</tr>
<tr>
<td></td>
<td>BT</td>
<td>Broader term, written as prefix to the superordinate term</td>
</tr>
<tr>
<td></td>
<td>NT</td>
<td>Narrower term, written as prefix to the subordinate term</td>
</tr>
<tr>
<td>associative</td>
<td>RT</td>
<td>these relationships can be used if equivalence or hierarchical relationships are not appropriate related term</td>
</tr>
<tr>
<td>Annotation</td>
<td>SN</td>
<td>A note attached to a term to indicate its meaning within the indexing language</td>
</tr>
</tbody>
</table>

Note that hierarchical relationships are semantically overloaded and covers three logically different situations: the generic relationship, the hierarchical whole-part relationship and the instance relationship.

3.2.1.2 ISO 5964:1985
ISO 5964 (1985) recommends documentation - guidelines for the establishment and development of multilingual thesauri, containing terms selected from more than one natural language. It displays not only the interrelationships between terms, but also equivalent terms in each of the languages covered.

The standard is focusing on the specific problems involved with multilingual thesauri:
- a term in a particular language can express a concept which cannot be represented by an exactly equivalent term. The standard summarizes five gradations from exact equivalence to non-equivalence (the target language does not contain a term which corresponds in meaning)
- management problems, e.g. a new thesaurus in another language can be translated from a source thesaurus or can be constructed ab initio.

The standard describes three main approaches to the construction of multilingual thesauri:
1. ab initio construction, i.e. establishment of a new thesaurus without direct reference to the terms or structure of any existing thesaurus.
2. translation of an existing thesaurus
3. merging of existing thesauri in two or more working languages

3.2.1.2.3 ANSI/NISO Z39.19:2003

NISO, The National Information Standards Organization, a non-profit association accredited by the American National Standards Institute (ANSI), the United States representative to the international standards organizations, has launched in 2003 an initiative to revise the leading standard for thesaurus construction: ANSI/NISO Z39.19, Guidelines for the Construction, Format, and Management of Monolingual Thesauri, the American equivalent of ISO2788. The Goal of the Revision is to Provide a revised version of Z39.19 that is relevant to the construction of thesauri, controlled vocabularies, taxonomies, and related groups of terminology.

NISO hopes that this revision addresses the needs of a changing information environment and a changing audience for this Standard. Searching and browsing of information systems are no longer limited to librarians, indexers, and other information professionals - individuals of all ages, professions, and nationalities use search tools for education, work and fun. NISO recognizes that developers of Internet and Intranet-accessible Web pages, databases, and information systems need better metadata to support non-expert information searches, and metadata developers are recognizing the value of incorporating high-quality, interoperable controlled vocabularies and taxonomies into their schemes.
3.2.1.3 Thesaurus and ontology: comparison
Both thesauri and ontology's are supporting access to documents, data and formal knowledge.
are suggesting that ontology's and thesauri could be integrated to manage the formal and informal knowledge in an organizational memory. They listed commonalities of thesauri and ontology's:
- Both include formal relations between concepts such as abstraction hierarchies, partonomies and definitions
- Both are designed to convey the intended meaning of concepts to humans, not only for use by computers
- Both are used to facilitate the retrieval of needed information from large amounts of available information
- Both are essential for translating between different conceptualizations and languages

The main difference between ontology's and thesauri, as described in ISO standard (1986b), is that a thesaurus is using generic standardized semantic relationships: the equivalence relations (USE), hierarchical relations (Narrow Term (NT), Broader Term (BT)) and associative relations (Related Terms (RT)). Using this kind of relationships, it is possible to build hierarchical structures or taxonomies. Ontology's don't have these limitations: any kind of relationships together with constraints could be defined. Another important difference is that ontology's within the Semantic Web are build using knowledge representation languages (OWL) with formal specified model theoretic semantics. For a more in-depth treatment of the meaning of semantics and model theoretic semantics see also section 3.3.

3.2.1.4 The Ontology spectrum
A widely cited definition of an ontology in the context of knowledge sharing is ‘A specification of a conceptualization’ That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general.

What is important is what an ontology is for. Ontology’s have been designed for the purpose of enabling knowledge sharing and reuse.

A wide variety of concepts are existing, attempting to address issues in the representation and classification of content and knowledge, enabling knowledge sharing and reuse: catalogues, glossaries, taxonomies, thesauri, conceptual models and logical theories. Therefore they could be considered as ontology’s, but they are not the same. They can be viewed as a spectrum of detail in their specification, visualized as a simple linear spectrum of definitions. A comparable spectrum of definitions is given by and a combination of both spectra is showed in Figure 2.
Figure 2: the ontology spectrum

According to, one of the simplest notions of a possible ontology is a controlled vocabulary, a finite list of terms. Another concept is a glossary, a list of terms and meaning. The meanings are specified typically as natural language statements.

Taxonomy may refer to either a hierarchical classification of things, or the principles underlying the classification. Almost anything—animate objects, inanimate objects, places, and events—may be classified according to some taxonomic scheme. Mathematically, a taxonomy is a tree structure of classifications for a given set of objects. At the top of this structure is a single classification—the root node—that applies to all objects (2004c). In a taxonomy, the semantics of a relationship between a parent and a child node is ill defined. In some cases, the relationship is the subclass of relation, in others it is the part of relation. In still others, it is undefined e.g. a computer directory structure.

Thesauri provide additional standardized relations (equivalence, hierarchical and associative relationships) between terms (see also section 3.2.1).

Conceptual models offer more expressive facilities for modeling and for structuring information bases. Languages supporting the construction of conceptual models provide semantic terms for modeling an application, such as entity and relationship as well as means for organizing information. Most conceptual models subscribe to an object-centered view of the world. Thus, their ontology includes notions like individual objects, which are associated with each other through (usually binary) arbitrary relationships, and which are grouped into classes.

Conceptual models build using formal languages based on logic, with formal model theoretic semantics, results in a logical theory. In a logic-based approach, the representation language is usually a variant of first-order predicate calculus. Automated reasoning is possible and amounts to verifying logical consequence. See also sections 3.4 about Description Logics and 3.3.3 about first order predicate logic.
3.2.1.5 Migration from thesauri to ontology’s
Semantic web technology claims to make it possible to integrate various information sources by the use of ontology’s. The domain model implicit in an ontology can be taken as a unifying structure for giving information a common representation and semantics. Such ontology representations are planned to fulfill the role currently undertaken by thesauri. Therefore a migration path is required from current thesauri to ontology’s, or support there co-existence if those ontology’s are to be adopted and assimilated into existing information retrieval infrastructure.

3.3 Semantics
In the previous paragraphs, the word ‘semantic’ is often used. What does this word mean in general and in the context of the semantic web?

From a linguistics point of view, according to , semantics is the study of the meaning of words, phrases, and sentences in language. It explores the minimum of knowledge about a linguistic sign or combinations of signs such that the expression can convey a specific communicative content.

Meaning as stated in this definition seems to be very strongly connected to the mind, to human perception. In section 3.3.1 it will be investigated if there is some possibility to represent (part of) meaning with other means besides the human brain. In section 3.3.1 the semantic continuum will be presented, showing different possible ways to specify semantics. In section 3.3.3 First Order Predicate Logic will be reviewed as this is an important knowledge representation formalism, forming the basis of Description Logics, where also OWL belongs to. Also the meaning of semantics within predicate logic will be described. In section 3.3.3.2 the possible meanings of semantic of the ‘semantic web’ will be indicated. A clear distinction between real world and model theoretic semantics will be made.

3.3.1 Mind-brain problem
The notion of semantics as stated above, seems to be something which is related to the mind of a person and the question if semantics could have a physical representation (e.g. a set of RDF assertions) is related with the mind-brain problem. This is in fact already for ages a key-problem in philosophy. There are innumerable formulations to this problem (John Beloff 1994). On one side, one is claiming that the mind is something totally different from physical bodies (the brain), and that we cannot explain what the one is in terms of the other at all. On the reverse side on is claiming that there is some way to explain what the mind, a mental substance, is in terms of a physical substance.
Research towards artificial intelligence has led to the Physical Symbol System Hypothesis, a conception that the mind is the result of a physical system.

The hypotheses states:

"A physical symbol system has the necessary and sufficient means for intelligent action."

This means that the fundamentals of our mind, and also meaning and semantics, could be incorporated in other media than human brains, e.g. machines.

A possible mapping from the mental world to the physical world is summarized in Table 2: mapping from mental world to physical world (Luc Steels 1992):

**Table 2: mapping from mental world to physical world**

<table>
<thead>
<tr>
<th>Mental world</th>
<th>Physical world</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts</td>
<td>Physical symbols</td>
</tr>
<tr>
<td>Conceptual structures</td>
<td>Structures of physical symbols</td>
</tr>
<tr>
<td>Conceptual embedding</td>
<td>Causal connections</td>
</tr>
<tr>
<td>Mental activity</td>
<td>Physical process between physical symbol structures</td>
</tr>
<tr>
<td>Learning</td>
<td>Physical process that changes the physical symbol structures</td>
</tr>
</tbody>
</table>

This means that concepts can be represented by unique physical objects or symbols of any kind (paper, bits, strings…). The physical objects are part of bigger structures, the context of the object. If we only represent part of the world with physical symbols, than we have a model of the world.

3.3.2 Description and specification of semantics

Semantics can be specified and described on many different ways. describes this with the semantic continuum, presented in Figure 3: the semantic continuum and explained further in this section.

**Figure 3: the semantic continuum**

When meaning is conveyed based on a shared understanding derived from human consensus, the semantics are not explicitly expressed outside the human brain. The semantics are implicit only.

An example is a set of XML tags, such as ‘author’ or ‘subject’. There is a consensus among the users of the XML document to interpret the tags in a specific way so that it can be hardwired in applications using this information. The risks of this approach is that people not always agree about the implicit meaning, leading to ambiguity.
The next step in the semantic continuum is an explicit but informal specification, meant for use by humans. At this moment, machines have limited abilities to make use of informally expressed semantics. An example of an explicit representation of informal semantics is a standard, e.g. ISO 5964:1985, the standard about thesauri, described in natural language. The expressed semantics could then be hard coded in working software.

It is also possible to express semantics formally, using a formal language, but still intended for human processing. Examples are mathematical descriptions of a model or the use of logic to express the processing of an electronic circuit. Formal semantics can prevent ambiguous interpretation but not completely.

The last step is a formal specification of semantics intended for machines for direct processing using automated inference. Processing of this semantics could be done on a procedural base. It is hardwired in a procedure what has to be done when a certain token is encountered. E.g. when in a compiler a ‘+’ is encountered, the appropriate procedure is carried out. The semantics is equivalent to the result of the procedure.

Another way is a declarative approach, using a logic based semantics, e.g. using first order predicate logic. Because OWL DL is equivalent to a Description Logic, a subset of first order predicate logic, the basics and meaning of semantics of first order predicate logic will be explained in the next section.

3.3.3 First order predicate logic

An important formal knowledge representation language is arguably predicate logic (or strictly, first order predicate logic - there are lots of other logics out there to distinguish between). Predicate logic allows to represent fairly complex facts about the world, and to derive new facts in a way that guarantees that, if the initial facts were true then so are the conclusions. It is a well understood formal language, with well-defined syntax, semantics and rules of inference.

It is not the intention of this section to define formally the syntax and semantics of predicate logic. A good introduction to this subject could be the book. Syntax and semantics of the first order logic will be explained informally because it is this notion of semantics that allows processing by machines.

3.3.3.1 Language constructions

Predicate logic allows to represent fairly complex facts because it has syntactic means to build sentences representing facts.

Sentences in predicate logic are built up from atomic sentences. Atomic sentences consist of a predicate name followed by a number of arguments. These arguments may be any term. Terms may be:
Constant symbols such as `bert'.

Variable symbols such as `X'.

Function expressions such as `father(bert)'. Function expressions consist of a function followed by a number of arguments, which can be arbitrary terms.

More complex sentences could then be constructed by using the *Boolean connectives* of the proposition logic: and, or, not, implication. *Quantifiers* are a means to express facts about a more specified amount of things e.g. the universal quantor and the existential quantor.

### 3.3.3.2 Predicate logic: semantics

An example of a predicate logic sentence is `R(x,y)'. It is possible that `R' means 'has a relation', indication that x has a relation with y. R could also mean 'has a horse' meaning that x has a horse y a totally different meaning.

It is clear that syntactical objects, e.g. predicate logic sentences, don't have any semantics. They have to be interpreted from case to case. For interpreting the semantics of predicate logic, there are three aspects involved: the formal language, an interpretation function and a situation or structure. A structure could be a real domain of interest or a mathematical space. With the formal language, one can make descriptions of a certain structure. The connection between language expressions and the structure is made by the interpretation function. An interpretation function defines the basic meanings (truth values) of the basic components, given some domain of objects. So, if the domain has only 4 objects (e.g. Peter, Bert, Marleen, Jan), it is possible to define the meaning of the predicate R in terms of all the pairs of objects for which the R relationship is true - say R(Bert, Marleen) = true.

Formally posed, a structure D is a triad D,R,O with D a non empty set, the domain of discourse. R a set of relations on D and O a set of operations on D. Predicates correspond syntactically with relations and operations with functions. An interpretation function I maps every individual c of a predicate logical language onto an object I(c), element of O. I maps also every predicate P onto a relation I(P) element of the set of relations R and every function f onto an operation I(f) element of the objects of D. A pair (D,I) is also named a model. There is also a function which maps every variable x onto an object b(x) element of D.

An important concept of a logical language is entailment. A set of sentences or formula's F entails another sentence or formula x if a model of F is also a model of x.

A theory of a model M is the set of sentences which are all true in M. A theory could also be defined by an explicit description of all true sentences also called axioms. This technique is often used to set up a knowledge base and is also called the model-theoretic semantics.
3.3.4 Semantics and the Semantics web
Two main views concerning the use of semantics for the Web are existing: one is what might be called the "human-meaningful" approach, and the other is the formal approach to semantics, exemplified by model-theoretic semantics.

and describes these kinds of semantics existing today on the web and of the emerging new semantic web infrastructure: real world, axiomatic and model-theoretic semantics. These concepts are further elaborated in subsequent paragraphs.

3.3.4.1 Real world semantics
Real world semantics are concerned with the “mapping of objects in the model or computational world onto the real world and issues that involve human interpretation, or meaning and use of data or information”.

Within the Semantic Web standards, URI’s, meant for identifying resources are used to refer to concepts. With RDF, the base assertional language of the Semantic Web, it is possible to relate various concepts and to build up conceptual structures. OWL adds more expressional power. Using the RDF(S) and OWL one can construct a possibly large set of expressions, which collectively are intended to represent some real world domain. With the upcoming standards implementing the logic layer, physical processes between physical symbol structures are possible. One could conclude that the Semantic Web is able to represent the mind and that “semantic” is related with the real world semantics. According to the idea of real world semantics, as described above captures the essence of the main use of the term “semantics” in a Semantic Web context.

Using only this view on semantics for the semantic web leads to problems: there is a potentially infinite set of human judgments about meanings of syntactical structures, RDF(S) and OWL assertions in the case of the semantic web.

3.3.4.2 Model theoretic semantics
A major goal of the Semantic Web is to have computers be able to interpret web mark-up language constructions unambiguously and make valid inferences about information resources, without having to involve people in the loop. Model theory, a basic technique for specifying the semantics of a formal language, is used for the definition of the semantics of Semantic Web languages (Patrick Hayes 2004).

3.3.4.2.1 Model theory
Model theory is a branch of mathematical logic that deals with 'truth in a structure', where by 'structure' is meant a set-theoretic structure, that is, a set together with particular relations and functions defined on it, along with certain constants elements distinguished in the set . It is also this way that the semantics of formal logic languages defined (see also section 3.3.3.2 where the semantics of first order logic are explained).

Model theory started with the study of formal languages and their interpretations but in a broader sense, model theory is the study of the
interpretation of any language, formal or natural, by means of set-theoretic structures, with Alfred Tarski’s truth definition as a paradigm.

3.3.4.2.2 Basic meaning of model theoretic semantics
Model theoretic semantics restricts itself to a formal notion of meaning which could be characterized as the part that is common to all other accounts of meaning, and can be captured in mechanical inference rules. Model-theoretic semantics is an "account of meaning in which sentences are interpreted in terms of a model of, or abstract formal structure representing, an actual or possible state of the world". A model theoretic semantics for a language assume that the language refers to a ‘world’, and describes the minimal conditions that a world must satisfy in order to assign an appropriate meaning for every expression in the language (Patrick Hayes 2004). The basic intuition of model-theoretic semantics is that asserting a sentence makes a claim about the world, it is another way of saying that the world is so arranged as to be an interpretation which makes the sentences true.

It is this kind of semantics which can be ‘understood’ and processed by machines.

3.3.4.2.3 Capability of model theoretic semantics
In the vision of Tim Berners Lee, the focus for the semantic web is on machine processable content. To accomplish this, machines should be able to:
- interpret unambiguously the legitimate expressions of a given language
- evaluate the truth of a language statement under a particular interpretation
- carry out automated reasoning with these statements

By using the model-theoretic approach, a formal abstract structure to interpret language statements is introduced. The interpretation and evaluation of the truth of a language statement is evaluated referring to this abstract structure. The structure can be encoded by a machine. The truth of a statement under a given interpretation can then be assessed mechanically. The important notion of entailment or satisfiability lies at the heart of logical reasoning as explained in section 3.3.3.2, and can be seen as a way to derive the truth of a statement from the truth value of other statements. Entailment is the formal counterpart to the intuitive notion of one statement derived from others. It deals with semantics (as opposed to syntax), because it is concerned with interpretations of expressions and with the truth of the expressions. Different algorithms are existing for automated calculation of satisfiability. The most recent knowledge representation systems based on Description Logics adopt tableau calculi.
3.3.4.2.4 Connection to real world semantics

Based on the previous paragraphs, one could conclude that model theoretic semantics permit machines to understand meanings in the ways that humans understand natural language. This is not the goal of using model theoretic semantics. According to (Patrick Hayes 2004), the chief utility of a formal semantic theory is not to provide any deep analysis of the nature of the things being described by the language but to provide a technical way to determine when inference processes are valid. The same counts for the interpretation of sentences in first-order logic, which were never intended to mean anything. Rather they were designed to express conditions which things can satisfy or fail to satisfy. Model theory is not a semantical theory which relates natural languages to the physical and social reality, but rather a mathematical theory which relates some mathematical structures to other mathematical structures. Model-theoretic semantics does not give meanings to natural-language expressions, but it can be considered as a way to describe meaning on a formal machine processable way.

3.3.4.3 Axiomatic semantics

The notion of axiomatic semantics is closely related to model theoretic semantics in a way that an axiomatic semantics for a language specifies a mapping of a set of descriptions in that language into a logical theory expressed in first order predicate calculus. An example of axiomatic semantics for the Semantic Web languages is the specification of a mapping of a set of descriptions in RDF, RDFS, DAML+OIL into a logical theory expressed by first-order predicate calculus. The basic idea is that the logical theory produced by the mapping is logically equivalent to the intended meaning of that set of descriptions. The mapping also produces a representation of the descriptions from which inferences can automatically be made using traditional automatic theorem provers and problem solvers.

3.3.5 Conclusion

Semantics, also within the context of the Semantic Web, has many possible meanings. A key to the successful operation of the Semantic Web with machine processable content is the use of a formal language for specifying the content. An author attempts then to communicate meaning by specifying axioms in a logical theory. This way an actual model is defined corresponding to what the author has represented (which is not necessarily what he wanted to represent by making the assertions). Semantics referring to the model-theoretic foundation of RDF(S) and OWL, is a much narrower notion then ‘real world semantics’. The virtue of the model-theoretic approach to semantics is that it focuses the conventions for dealing with meanings onto a consolidated framework that provides the rules for assigning well-defined interpretations to language expressions, as well as a mechanism for determining when one statement.
is entailed by other statements. Such a framework can be used to perform automated reasoning over the Web.
3.4 Description logics

The Semantic Web ontology languages and more specific OWL DL are strongly related to description logics. In this section Description Logics will be explained.

3.4.1 Definition and history

Description Logics are a formalism for representing knowledge, as well as some important basic notations underlying all systems that have been created in the DL tradition.

Approaches to knowledge representation developed in the 1970’s are sometimes divided roughly into two categories: logic based formalism and non-logic-based representations.

In a logic based approach, the representation language is usually a variant of first-order predicate calculus, and reasoning amounts to verifying logical consequence or entailment.

In non-logic based approaches, knowledge is represented by means of ad hoc data structures and reasoning is accomplished by ad hoc procedures that manipulate the structures. An example are semantic networks and frames.

Network based systems, were often considered more appealing and more effective from a practical viewpoint than logical systems but a problem was the lack of precise semantic characterization. Every system behaved differently despite their virtually identical looking components. An important step in further development was the recognition that frames could be given a semantics by relying on first-order logic. Indeed, further research turned out that frames and semantic networks could be regarded as fragments of first order logic. In more recent years, after attention was further moved towards the properties of the underlying logical systems, the term Description Logics became popular.

3.4.2 Knowledge representation in Description Logics

Knowledge can be described by means of ontology’s which can be described with the aid of Description Logics.

The realization of knowledge systems involves two primary aspects. The first consist of providing a precise characterization of a knowledge base. This involves the precise characterization of the type of knowledge to be specified to the system as well as clearly defining the reasoning services the system needs to provide, the kind of questions the system should be able to answer. The second aspect consist of providing an environment where the user can benefit from different services that can make his interaction with the system more effective.

Within a knowledge base, there is a clear distinction between intentional knowledge or general knowledge about the problem domain and extensional knowledge, which specifies to a particular problem. A DL knowledge base is comprised by two components, a T-box and a A-box. The T-box contains intentional knowledge in de form of a terminology and
is built through declarations that describe general properties of the concepts. The A-box contains extensional knowledge, also called assertional knowledge, knowledge that is specific to the individuals of the domain of discourse. Two types of assertions are existing: concept assertions and role assertions.

3.4.3 Description logics semantics
The standard technique for specifying the meaning of a Description Logic is via a model theoretic semantics, whose purpose is to explicate the relationship between the language syntax and the intended model(s) of the domain. A model consists of a domain (often written $I$) and an interpretation function (often written $\cdot I$), where the domain is a set of objects and the interpretation function is a mapping from individual, class and property names to elements of the domain, subsets of the domain and binary relations on the domain, respectively.

For a treatment of model theoretic semantics see also section 3.3.4.2.

3.4.4 From ALC to SHIQ
The Description Logic ALC, which was first described by Schmidt-Schauss and Smolka, is the ‘smallest’ DL that is propositionally closed, i.e., that provides for all Boolean connectives. More precisely, ALC concepts are build from the Boolean connectives and so-called existential and universal value restrictions. (Carsten Lutz 2002)

Figure 4 summarizes the different ALC constructors together with their names and semantics. The semantics are defined on a model theoretic way, by the interpretation $I = \langle D, \cdot I \rangle$ with $D$ a non empty set, the domain and $\cdot I$ an interpretation function which maps every concept on a subset of $D$ and every role on the power set $D \times D$. Every individual object is mapped on an element of $D$. An interpretation functions is also an extension function if it satisfies the semantic definitions of the language.

**Figure 4**

<table>
<thead>
<tr>
<th>$A$</th>
<th>$\forall x. A^I \subseteq \Delta^I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R$</td>
<td>$\exists x. \forall y. R^I(x,y) \rightarrow C^I(y)$</td>
</tr>
<tr>
<td>$\top$</td>
<td>$\Delta^I$</td>
</tr>
<tr>
<td>$\bot$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$\neg C$</td>
<td>$\Delta^I \setminus C^I$</td>
</tr>
<tr>
<td>$C \sqcap D$</td>
<td>$C^I \cap D^I$</td>
</tr>
<tr>
<td>$C \sqcup D$</td>
<td>$C^I \cup D^I$</td>
</tr>
<tr>
<td>$\forall R. C$</td>
<td>${ x \mid \forall y. R^I(x, y) \rightarrow C^I(y) }$</td>
</tr>
<tr>
<td>$\exists R. C$</td>
<td>${ x \mid \exists y. R^I(x, y) \land C^I(y) }$</td>
</tr>
</tbody>
</table>

primitive concept
primitive role
top
bottom
complement
conjunction
disjunction
universal quant.
existent quant.
Several extensions have been proposed to increase the expressive power of ALC:

- ALC extended with transitive roles named ALC R+ or S
- S extended with role hierarchies named SH
- SH extended with inverse roles named SHI
- SHI extended with number restriction named SHIQ

The most important constructors together with their semantics are summarized in figure 5.

### figure 5

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Syntax</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>concept name</td>
<td>$A$</td>
<td>$A^I \subseteq \Delta^I$</td>
</tr>
<tr>
<td>top</td>
<td>$\top$</td>
<td>$\Delta^I$</td>
</tr>
<tr>
<td>bottom</td>
<td>$\bot$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>conjunction</td>
<td>$C \sqcap D$</td>
<td>$C^I \cap D^I$</td>
</tr>
<tr>
<td>disjunction</td>
<td>$C \sqcup D$</td>
<td>$C^I \cup D^I$</td>
</tr>
<tr>
<td>negation</td>
<td>$\neg C$</td>
<td>$\Delta^I \setminus C^I$</td>
</tr>
<tr>
<td>universal</td>
<td>$\forall R.C$</td>
<td>${ x \mid \forall y : R^I(x, y) \rightarrow C^I(y) }$</td>
</tr>
<tr>
<td>existential</td>
<td>$\exists R.C$</td>
<td>${ x \mid \exists y : R^I(x, y) \land C^I(y) }$</td>
</tr>
<tr>
<td>cardinality</td>
<td>$\geq n R$</td>
<td>${ x \mid | y : R^I(x, y) | \geq n }$</td>
</tr>
<tr>
<td></td>
<td>$\leq n R$</td>
<td>${ x \mid | y : R^I(x, y) | \leq n }$</td>
</tr>
<tr>
<td>qualified cardinality</td>
<td>$\geq n R.C$</td>
<td>${ x \mid | y : R^I(x, y) \land C^I(y) | \geq n }$</td>
</tr>
<tr>
<td></td>
<td>$\leq n R.C$</td>
<td>${ x \mid | y : R^I(x, y) \land C^I(y) | \leq n }$</td>
</tr>
<tr>
<td>enumeration</td>
<td>${a_1 \ldots a_n}$</td>
<td>${a_1^I, \ldots, a_n^I}$</td>
</tr>
<tr>
<td>selection</td>
<td>$f : C$</td>
<td>${ x \in \text{Dom}(f^I) \mid C^I(f^I(x)) }$</td>
</tr>
</tbody>
</table>

### 3.4.5 Reasoning

A knowledge system has to provide reasoning services both on the T-box and the A-box.

The basic inference on concept expressions in Description Logics is the subsumption written as $C \subseteq D$. With subsumption it is possible to structure concepts of a knowledge base. This way it is possible to calculate a taxonomy of concepts and to automatically classify new concepts into the taxonomy.

Another reasoning service is concept satisfiability. Concept satisfiability checks if there are instances which satisfy the concept so that the concept is not empty. The algorithms for concept satisfiability and subsumption are often based on tableau calculus.

Two important properties are soundness and completeness of the algorithms used. Also the computational complexity of the algorithms is an important property. There is a tradeoff between expressiveness of a
representation language and the difficulty of reasoning over the language. In other words, the more expressive the language is, the harder the reasoning. That’s the reason why most DL’s are in fact a small subset of First Order Logic in order to deliver complete and sound reasoning services on a limited time.

3.4.6 Description Logics and the Semantic Web

OWL, the ontology language for the Semantic Web, developed by the World Wide Web Consortium (W3C) Web Ontology Working Group, was not designed in a vacuum. There were many influences on OWL’s design: it had to fit into the Semantic Web vision of a stack of languages including XML and RDF. OWL takes the basic fact-stating ability of RDF and the class- and property-structuring capabilities of RDF Schema and extends them in important ways.

OWL had to be able to represent a useful group of ontology features. Some of the most important influences on the design of OWL came, via its predecessor DAML+OIL, from Description Logics, particularly on the formalisation of the semantics, the choice of language constructors, and the integration of data types and data values. OWL comes in three flavours. In fact OWL DL and OWL Lite (two of the three species of OWL) can be viewed as expressive Description Logics, with an ontology being equivalent to a Description Logic knowledge base. A key feature of Description Logics is that they are logics, i.e., formal languages with well defined semantics. This is also the case for OWL. This was recognised as an essential feature, as it allows ontology’s to be shared and exchanged without disputes.

Another goal of the design process was to ensure that OWL entailment would at least be decidable, i.e., that it would be possible to design an algorithm that could guarantee to determine whether or not one OWL ontology entails another (such an algorithm is often called a decision procedure). A suitable balance between these computational requirements and the expressive requirements identified in was achieved by basing the design of OWL on the SH family of Description Logics. Members of the SH family include the SHIQ Description Logic [23], which adds inverse properties and generalised cardinality restrictions compared to SH, and SHOQ(D) [22], which adds the ability to define a class by enumerating its instances.

OWL Lite is similar to the Description Logic SHIF. Like SHIF, key inferences in OWL Lite can be computed in worst case exponential time (ExpTime), and there are already several optimized reasoners for logics equivalent to OWL Lite.

OWL DL—the Description Logic style of using OWL—is very close to the SHOIN(D) Description Logic which is itself an extension of the SHOQ(D) Description Logic (extended with inverse roles and restricted
to unqualified number restrictions). SHOIN(D) is a very expressive Description Logic. With this Description Logic it is possible to build complex Boolean descriptions using, for example, union and complement. SHOIN(D) is also difficult to reason with, as key inference problems have NExpTime complexity.

4 Related work

Some recent research is done to investigate the possibility to convert thesauri to ontology’s; the accent was on representation formalisms for converted thesauri.

4.1 Semantic Web representation formalisms for thesauri

published a proposal for a RDF representation of various conceptual relationships typical of controlled vocabularies such as thesauri, classification systems and organized meta data collections. The aim is to explore the use of RDF as a common formalism for representing a variety of different thesauri and classification systems within the same overall framework. By doing so, they expect to leverage generic RDF facilities (such as query and storage software components), and also to have a basis for mapping between subject classifications expressed using these various vocabularies. The proposal is still in draft, they identified four open issues.

investigated the way to convert an antique furniture thesaurus to an ontology. They discussed the representation requirements for such an ontology as well as representational problems for the sample ontology with respect to the emerging web standards for knowledge representation (RDF, RDFS, OIL). In a case study they used the furniture ontology as an annotation tool for describing antique furniture.

proposed a Thesaurus Interchange Format (TIF) in RDF. They represented a smaller (ca. 2000 terms) multilingual thesaurus (ELSST - Social Science Thesaurus) with the proposed TIF schema to test its coverage. No further details on the results and set-up of these tests were given.
A review of the current state of migrating from thesauri to ontology’s has been presented by (Michael Wilson 2003). He presented 3 different approaches to represent thesauri using semantic web standards and mentioned only the furniture thesaurus and the ELSST thesaurus as examples.

Within the Semantic Web Advanced Development (SWAD) for Europe work plan, work package 8 was defined very recently. The SWAD-Europe project aims to support W3C’s Semantic Web initiative, providing targeted research, demonstrations and outreach to ensure Semantic Web technologies move into the mainstream of networked computing.

The goal of work package 8, is the development of one or more research prototype RDF thesauri showing support for advanced characteristics such as ISO-compatibility, multi-linguality, relations to RDF ontology’s, classification schemes and cross-mapping between thesauri. Deliverable 8.2 of work package 8 (Alistair Miles and Brian Matthews 2003), is a review of the current thesaurus work. The result of this work is very recently published as a W3C Editor’s Working Draft under the name ‘SKOS Core Guide’.

SKOS Core provides a model for expressing the basic structure and content of concept schemes (thesauri, classification schemes, subject heading lists, taxonomies, terminologies, glossaries and other types of controlled vocabulary). The SKOS Core Vocabulary is an application of the Resource Description Framework (RDF), that can be used to express a concept scheme as an RDF graph. Using RDF allows data to be linked to and/or merged with other RDF data by semantic web applications.

There is a lot of variability in the schemas reviewed here. This variability includes some fundamental differences in the underlying data model that is implied. It also includes significant differences in how RDF has been used to represent the same information. At the very least, it must be understood how they relate to each other. However, many issues remain to be considered. For example, although there is standardization in this area, in practice thesauri come in many different flavors. A useful schema must be able to accommodate these subtle variations of meaning. Also, when it becomes a part of the semantic web, thesaurus data will be used alongside ontology’s expressed in languages such as OWL and many other types of knowledge organization scheme. A well designed schema would enable thesaurus data to be fitted seamlessly in to this semantic web.

A list of 10 open design issues still remains (2003c).

4.2 Conversion methods

Recently, also some work has been published describing methods for converting existing source material to a representation that is compatible with Semantic Web languages such as RDF(S) and OWL. The work of
focuses on a generic method to convert existing resources of various kind to a Semantic Web compatible representation, without altering the original material and semantics of the original representation. The presented method has two stages: a syntactic conversion stage and a semantic transformation stage.

The syntactic conversion stage contains two steps: a structure preserving syntactic translation from the source format to RDF(S) and the explicit representation of information that is implicit in the original data format. In the semantic conversion stage the RDF(S) instances generated in the syntactic stage are augmented according to the intended semantic of the source model. Also the model is reinterpreted in terms of the RDFS or OWL semantics. Were possible, build in RDFS/OWL properties must be used.

In this method is worked out further and is focused on ‘existing thesauri and related resources’. Two additional steps are identified: a preparation step and a standardization step.

During the fisrt step, the preparation step, an analysis of the thesaurus is made using it’s documentation, it’s digital form and/or by contacting the thesaurus authors. An analysis of the thesaurus contains the following:

- Conceptual model (the model behind the thesaurus is used as background) knowledge in creating a sanctioned conversion
- Relation between conceptual and digital model
- Relation to standards (aids in understanding the conceptual and digital model)
- Identification of multilinguality issues

Step two is a syntactic conversion step. Typical source representations such as proprietary text formats, relational databases and XML representations are converted from the source representation to RDF(S). recommends to carry out this step in two phases. The first phase is a structure preserving translation between the source format and RDF format. The goal is that the translation should reflect the source structure as closely as possible. The translation should be complete, meaning that all semantically relevant elements in the source are translated into RDF. The second phase is used to make information explicit that is implicit available in the source representation.

In step 3, named by the authors as semantic conversion step, the class and property definitions are augmented with additional RDF(S) and OWL constraints. For example, a broaderTerm property can be defined as an owl:TransitiveProperty and a relatedTerm property as an owl:SymmetricProperty. Also some specific interpretations are introduced in this step that are strictly speaking not sanctioned by the original model or documentation for application specific requirements.
The last step (step 4) is meant for mapping the thesaurus schema onto a *standard schema*. Note that at the moment a standard schema has not yet been agreed upon.
5 Proposal of a method for converting thesauri to RDFS/OWL

5.1 Introduction
In this work an adapted stepwise method is proposed to convert the existing thesauri to an RDFS/OWL representation, based on the method of . The goal is that the RDF(S)/OWL based thesaurus system can provide the same services as the actual thesaurus systems. See also chapter 8 for a list of thesaurus services. The scope of this work is limited to thesauri based on the ISO standards. But thesauri often deviates more or less from the standards, the RDF(S)/OWL system must be able to handle these variations.

Basic assumption of this method is that there exists an extendible standard (at least a standard for the organization using the thesauri) RDF(S)/OWL meta-model for thesauri. Such a model will be discussed and proposed in chapter 7.

5.2 Overview of the conversion method
The first step of the method is an analysis of the existing thesaurus system. Both the existing thesaurus meta model and the thesaurus data must be analyzed to be able to carry out the conversion correctly. In general, an automatic conversion of legacy thesaurus systems is not possible because the semantics of the meta-model and thesaurus data structures are not described on a formal machine processable way. Based on this analysis, a mapping can be defined between the source thesaurus meta model and the standard RDF(S)/OWL thesaurus meta model. When needed one has also to extend the standard RDF(S)/OWL thesaurus meta model. This mapping has to be done without altering the semantics (intended or specified) of the source model. Therefore this step is named ‘Semantic mapping’.

In a last step the source thesaurus data have to be translated into simple RDF assertions, using the terminology defined by the RDF(S)/OWL meta model. This is a purely syntactical conversion. The RDF assertions together with the RDF(S)/OWL meta model contains all data and semantics which can be interpreted and processed by a generic OWL reasoner to deliver the desired services.

See Figure 6 for a schematic overview of the method. This method is further explained in subsequent paragraphs.

Figure 6
5.3 Step 1: Analysis
In this step, an analysis of the existing thesauri is made using the existing documentation but certainly also by consulting the thesaurus authors and system administrators.
It is necessary to understand both the semantics of the thesaurus meta model and the structure of the thesaurus data but also the application specific semantics.
In case of the Agfa thesauri, the thesaurus meta model syntax and semantics of the Basis thesaurus constructions are described in the Basis manual by means of natural language. Knowledge of the thesaurus meta model proposed by the ISO standards is also necessary (see also chapter 6). Using this information, a mapping between the standard ISO thesaurus construction (relations between terms) and the Basis relations are presented in section 6.2.1. A description of the Agfa thesauri and semantics of the relationships are given in section 6.2.2. Also by means natural language descriptions.
Also a correct understanding is necessary of the source thesaurus data structures to be able to translate this data structures correctly to another format.

5.4 Step 2: extension of the standard thesaurus meta model (semantic mapping)
During the two last steps of the method presented by an already converted RDF(S)/OWL thesaurus meta model is mapped onto a standard RDF(S)/OWL thesaurus meta model eventually extended with application specific properties. From a content point of view, these activities are related with the activities of the first step, the preparation step where an analysis of the structure and the semantics of the source thesaurus meta-model is performed. Therefore, in the method presented here, the first actions after the preparation step is to extend the standard thesaurus meta model e.g. to add non standard relationships between terms or concepts to the extendible RDF(S)/OWL meta-model for thesauri. By defining them as sub properties of the existing generic thesaurus properties (hierarchical relation, equivalence relation, associative relation), a generic thesaurus service agent can always retrieve the terms related to these non-standard relationships. For making use of the specific semantics of these relationships, the service software needs to be adapted. For a proposal of generic thesaurus services see chapter 8. It is always possible to define them as sub properties of the generic thesaurus relations because they cover all possible types of relationships between terms or concepts, see also chapter 6.
5.5 Step 3: syntactic conversion of thesaurus data

proposes a structure preserving translation of the source format directly into an RDF format. In this work another approach has been chosen: in step 3a the source format is translated into a simple XML representation, translating the original syntax into a comparable XML format. A dump of the thesaurus in text format is used as source for conversion to XML as this is a general available format. This approach has some benefits:

- Programming a conversion from a text format into a XML format is more straightforward than a direct translation into RDF(S). XML Mark-up tags can be chosen freely to indicate the type of information in the intermediate XML format. A tree structure, the XML structuring mechanism, can be applied to preserve the structuring of the original format while with RDF(S) it is only triplets are possible. This way a comparable structure with the source thesaurus data structure can be build up in XML syntax.

- No unwanted semantics can sneak into the XML format at this stage, because XML tags do not have specified semantics, which is not the case for a RDF(S) representation.

In step 3B, the simple XML format is translated into the final RDF(S)/OWL format using the terminology of the adapted standard RD(S)/OWL meta model.

In Figure 7 an overview is given of the syntactic conversion together with a sample of the thesaurus data for each step.
5.5.1 Step 3a: conversion to an XML representation

The syntactic conversion of the DE thesaurus was carried-out this way. The DE thesaurus was extracted from the Basis system as a text file and compiled with a simple Python program into an XML format (Python was chosen because of its text processing capabilities). See Figure 8: text output from the DE thesaurus and Figure 9: intermediate XML file.

Figure 8: text output from the DE thesaurus

```
AROXYDE
UF 3079
ARO-
AROXYDE
ARYL OXIDE
ARYL OXIDE
ARYL OXIDE
ARYL OXIDE
BT 2-PHENYLPHENOLATE SODIUM SALT
PHENYL OXIDE
PHENOXIDE
SN ARYL OXIDE ION CONTAINING COMPOUND; LINK WITH THE APPROPRIATE SALT
```

Figure 7
Figure 9: intermediate XML file

```
<term>AROXIDE</term>
<uf>3070</uf>
<uf>AR</uf>
<uf>AROXIDE</uf>
<uf>ARYL OXIDE</uf>
<uf>ARYL OXYDE</uf>
<uf>ARYLOXIDE</uf>
<uf>ARYLXYDE</uf>
<uf>2-PHENYLPHENOLATE SODIUM SALT</uf>
<uf>NAPHTHYL OXIDE</uf>
<uf>PHEXIDE</uf>
<uf>ARYL OXIDE CONTAINING COMPOUND LINK WITH THE APPROPRIATE SALT</uf>
```

The relation identifiers of the Basis format (UF, NT, SN in Figure 9: intermediate XML file) were converted to similar XML tags. The structuring tokens of the Basis text format are blancs and the use of the line-feed and end-of-line characters. This is not possible in XML, additional tags have to be used to preserve the structure and the implied semantics of the original representation.

The Basis text file contains several blocks, separated by empty lines. In one block, a term is related to other terms. The start of a new block in the Basis file is indicated by the presence of a string starting at position 1 of the line, representing a term, after a line feed. The end of the block is indicated by a line feed. The start of a new block in the XML representation is indicated by a tag `<term>` on level 1, the end of the block by a `</term>` tag on the same level. The term itself is tagged in XML by a `<term></term>` tag on level 2.

Relations are represented in the text file by the strings starting at position 3 of a line, if position 1 and 2 are empty. The terms on which the relations applies are starting on position 9 of this and all subsequent lines until a new relation is given on position 3 or the block ends. This structuring is translated into XML by surrounding the terms starting at position 9 with tags indicating the relation (the same relation identifiers are used as in the Basis file).

See appendix 11.1 for a dump of the Python source code.

This shows that knowledge of the semantics of the original text format is needed to make a complete and correct syntactical translation, but these semantics are preserved and unchanged in the new representation. Note also that there are different possibilities to structure the new XML representation. The criterion taken under consideration here was the ease of programming of the translation.

5.5.2 Step 3B: conversion to RDF triples

In step 3B, the simple XML format is translated into the final RDF(S)/OWL format using the terminology of the adapted standard RD(S)/OWL meta model. In this work a proposal is made for such a thesaurus model. See chapter 7: Modeling thesauri using Semantic Web languages.
Having such a standard semantic web compatible thesaurus model is important for various reasons:

- it can help the interchange of thesaurus data
- it will save a lot of time by converting existing thesauri to the web because
- a generic thesaurus services module could then be used to actually interpret the thesaurus. What generic thesaurus services are and how they could be delivered is explained in chapters 8 and 8.5.

If a standard thesaurus model is present and described in RDFS/OWL, then the work in this step is reduced to defining a mapping between the XML tags and elements of the intermediate XML file of step 2a and the appropriate RDF triples, as defined in the standard RDF(S)/OWL model and the execution of the translation.

A possible way to make a specification for translation of the intermediate thesaurus XML file to RDF(S)/OWL is by using XSLT (James Clark 2004). For the execution of the translation a XSLT processor could be used e.g. Saxon (Michael Kay 2004). XSLT is the abbreviation for Extensible Style sheet Language Transformation and is a XML transformation language, which transforms documents in XML format. The language is declarative, i.e. a program consist of a collection of several rules which transformations should be performed. The rules are applied recursively.

The XSLT processor checks which rules can be applied and executes the associated transformations based on a sequence of priorities. By using XSLT only the transformations rules must be designed not the actual translation algorithms.

The transformation from the simple XML thesaurus representation to the standard RDF(S)/OWL XML syntax has been worked out as a set of XSLT rules. A possible specification for translation is presented in Figure 10: XSLT for translation of the thesaurus in XML format to XML/RDF. This translation specification ties in with the RDF(S)/OWL thesaurus model worked out in chapter 7.

It turned out that using XSLT for this purpose is very convenient: once familiar with the XSLT syntax, such a set of transformation rules can be build quickly, is more easy to read in comparison with an imperative program, changes and additions simple to implement.

To carry out the conversion, Saxon was used.
Figure 10: XSLT for translation of the thesaurus in XML format to XML/RDF

5.6 Discussion

The proposed method to convert existing thesauri to a set of RDF assertions with semantics defined by an extended standard RDF(S)/OWL meta model is based on previous proposed methods. The main difference with the method of and is that a final thesaurus meta model is constructed first starting from a generic thesaurus RDF(S)/OWL
meta model. Due to the model theoretic foundation of the RDF(S)/OWL constructions this meta model can be interpreted by a computer.

In the method of this is carried out as the last step, just before publication and use of the thesaurus and is named the ‘Semantic conversion step’. While it is in principle unimportant when this step is carried out, either at the beginning or as the last step, from a content point of view, the activities of this step are connected with the activities of the first step, the preparation step where an analysis of the structure and the semantics of the source thesaurus meta-model is performed. The actual goal is to specify the RDF(S)/OWL meta-model as accurate as possible in order to allow generic reasoners to deliver the desired services, based on the model theoretic semantics of the RDF(S)/OWL constructions. A correct understanding of the semantics of the source thesaurus meta model is necessary and with this knowledge, the standard RDF(S)/OWL meta model can be extended accurately. This is further described in section 7.5. The term ‘Semantic conversion’ seems to be inaccurate. The semantics of the source meta-model are not converted but the standard RDF(S)/OWL thesaurus meta model is adapted in a way that the underlying model theoretic semantics correspond to the semantics of the source thesaurus system. Therefore the term ‘Semantic mapping’ is a more accurate description of this activity.

The further steps are then pure syntactical conversions of the actual thesaurus data. For efficiency reasons this is implemented in two steps. In a first step (step 3a) an XML format is generated from the source data using a simple imperative computer program. The resulting data structure is kept as similar as possible as the data structure of the source thesaurus. Having a XML representation, XSLT can be used to further generate the RDF triples containing the thesaurus data. Of course it is possible to do this using only on step by making a program that produces the RDF triples directly. But this requires a more complex program because the data structure of a set of RDF triples differs more form the source thesaurus data structure. A set of RDF triples contains less structure because the semantics are now described mainly by the RDF(S)/OWL meta model while in the case of the source thesaurus the semantics must be derived from the data structure. Using XSLT declarations is a convenient way to carry out the conversion because they contain only the conversion rules and not the entire conversion logic resulting in a clearer conversion description easier to debug or adapt compared to an imperative computer program.
6 Agfa Thesauri meta model

6.1 Introduction

In order to develop a migration path from an existing thesaurus to a web representation with web ontology languages, it is necessary to have a meta model of the thesaurus and a clear understanding of the semantics of this model, either explicitly specified or build-in in procedures used to deliver the thesaurus services.

The structure of thesauri are controlled by international standards, ISO 2788 and ISO 5964. The RDM thesauri are constructed in compliance with the ISO 2788:1986 standard.

The meta model and semantics of thesauri as defined in ISO 2788 and ISO 5964 are reviewed in section 3.2.1.2. In this chapter the meta model of the Agfa thesauri will be made described and analyzed using natural language.

This chapter presents the results of the first step, the preparation, of the conversion method described in broad outline in chapter 5.

6.2 Agfa thesauri

Agfa has developed during a period of 25 years several thesauri, over 1.500.000 terms. The most important are:

- INVENT.IN, containing product identifications
- INVENT.DE, containing document descriptors
- INVENT.PF, containing project identifications
- SECURITY, containing risk and safety (R&S) phrases in different languages
- TELEF.REGIST, containing person identifications

The thesauri are part of the BASIS document management system and managed by the Basis toolkit BASIS/TM (Thesaurus Manager). The underlying thesaurus meta model of BASIS/TM will be discussed first. In the next sections, the structure of the Agfa thesauri will be analyzed.

6.2.1 BASIS/TM data model

BASIS/TM is described in (2002). The thesaurus is a list of terms with codes, called relation types that indicate the type of relationships between terms. The different relation types together with their meaning and usage are:

- PREFER: Points to the preferred term in a relation.
- NONPREFER: Points to the non-preferred term in a relation.
- GENERAL: Points to a term that conveys a broader or more general concept.
- SPECIFIC: Points to a term that conveys a narrower or more specific concept.
COMPONENT: Points to one of several different terms that make up a concept.

COMPOSITE: Points to a term that conveys several different concepts.

REFERENCE: Points to an indirectly related term.

INFORMATIVE: Lets you write a message that gives the scope, history or other information about a term.

The name of a relation with a specific relation type can be anything.

Note that Prefer and Nonprefer, General and Specific, Component en Composite are reciprocal relations. Such a relation has to be defined only in one way. The reciprocal relation will be added automatically by the system.

Based on this information, it is possible to define a mapping between BASIS/TM relation types and ISO relations (classes). The mapping is presented in Table 3.

Note that COMPONENT and COMPOSITE are specific non standard relation types to point to the composing parts of a term or to a composite term of a simple term.

**Table 3: mapping between ISO and BASIS/TM thesaurus relations**

<table>
<thead>
<tr>
<th>ISO Class</th>
<th>ISO abrev.</th>
<th>Description</th>
<th>Basis/TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>equivalence</td>
<td></td>
<td>relationship between preferred and non-preferred terms.</td>
<td></td>
</tr>
<tr>
<td>USE</td>
<td></td>
<td>Use, Prefix to the preferred term</td>
<td>PREFER</td>
</tr>
<tr>
<td>UF</td>
<td></td>
<td>Use for, Prefix to the non-preferred term</td>
<td>NONPREFER</td>
</tr>
<tr>
<td>hierarchical</td>
<td></td>
<td>Providing a structuring mechanism based on the degree of super ordination and subordination</td>
<td></td>
</tr>
<tr>
<td>TT</td>
<td></td>
<td>Top term in a hierarchy</td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td></td>
<td>Broader term, written as prefix to the super ordinate term</td>
<td>GENERAL</td>
</tr>
<tr>
<td>NT</td>
<td></td>
<td>Narrower term, written as prefix to the subordinate term</td>
<td>SPECIFIC</td>
</tr>
<tr>
<td>associative</td>
<td></td>
<td>these relationships can be used if equivalence or hierarchical relationships are not appropriate</td>
<td></td>
</tr>
<tr>
<td>RT</td>
<td></td>
<td>A note attached to a term to indicate its meaning within the indexing language</td>
<td>REFERENCE INFORMATIVE</td>
</tr>
<tr>
<td>(Annotation)</td>
<td>SN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.2.2 Analysis of Agfa Thesauri

In the following sub paragraphs, an analysis is made of the Agfa thesauri. The goal is to have clear view on the data structure and semantics (also real world semantics) of the constructions.

6.2.2.1 INVENT.IN

6.2.2.1.1 Concepts

INVENT.IN contains concrete and abstract entities. Concrete entities are materials. Materials in INVENT.IN are of different nature: chemical compounds, photographic films, … Every materials is identified by a unique string, the lead term in the thesaurus (LT). Abstract entities are roles. They are used to indicate the role or use of a material in a particular context.

6.2.2.1.2 Equivalence relations

Often, materials have alternative identifications, such as commercial names, ISI-ISIM-INTM-ISIF numbers, Casnr. (Chemical Abstracts number), other Agfa internal identifications typical for a certain department. The lead term is connected to this alternative id’s by the UF relation. Synonyms of roles are also connected to the LT by the UF relation.

6.2.2.1.3 Hierarchical relations

In the case of chemical compounds, the NT relation is used for relating LT’s to products with the same chemical structure but different qualities (solution, paste, dispersion, …). If a chemical is a solution, then the scope note of the term contains ‘oplossing’. If a chemical is a dispersion, the scope note of the term contains ‘dispersie’.

The RT relationship means many different things:
- indication of a tautomeric compound
- water free compounds are connected to there hydrates by the RF relation
- compounds are related to there isomers by the RF relation
- cyclic compounds are related to there non cyclic structures by the RF relation
- a compound is related to a polymer loaded with the compound by the RT relation

In the case of roles, the NT and BT relationships are used to build a hierarchical structure of roles mostly of a generic type.

6.2.2.2 INVENT.DE

6.2.2.2.1 Concepts

INVENT.DE is the most important thesaurus and contains document descriptors. Within INVENT.DE several classification trees are existing e.g. for dyes, sensitizers, analytical techniques, polymers…
Also compound descriptors are used.
Some existing terms are showed in Figure 11.

**Figure 11**

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESENSITIZATION AGENT</td>
</tr>
<tr>
<td>DESENSITIZATION DYE</td>
</tr>
<tr>
<td>DESENSITIZATION SPECTRAL SENSITIZER</td>
</tr>
<tr>
<td>DESENSITIZER</td>
</tr>
<tr>
<td>DESENSITIZER DYE</td>
</tr>
</tbody>
</table>

6.2.2.2 Equivalence relations
The equivalence relationship is used for the indication of synonyms, sometimes also for linking to translations in another language. The name of the relationship is then adapted. E.g. for a German term, the equivalence relation UD is used.

6.2.2.3 Hierarchical relations
BT and NT relations are used for building up the hierarchy of descriptors. The hierarchical relations are of a generic type. An example is presented in Bout: Bron van verwijzing niet gevonden.

**Figure 12**

<table>
<thead>
<tr>
<th>Termen</th>
<th>Aandl</th>
<th>Related Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>DESENSITIZER DYE</td>
<td>11633 (UF)</td>
</tr>
<tr>
<td>127 128</td>
<td>DESENSITIZER (BT)</td>
<td>2000 2000 DYE (BT)</td>
</tr>
<tr>
<td>1 1</td>
<td>GREEN SPECTRAL DESSENSITIZER (NT)</td>
<td>RED SPECTRAL DESSENSITIZER (NT)</td>
</tr>
<tr>
<td>0 0</td>
<td>INFRARED SPECTRAL DESSENSITIZER (NT)</td>
<td></td>
</tr>
</tbody>
</table>

6.2.2.4 Associative relations
The RT relation is mainly used to link compound terms to the constituent parts. or to cover associations that are neither hierarchical nor equivalent.

6.2.2.3 INVENT.PF

6.2.2.3.1 Concepts
INVENT.PF contains individual entities: projects. The LT contains the actual project identification.

6.2.2.3.2 Equivalence relations
The LT is connected to alternative and expired project identifications by the UF relation.

6.2.2.3.3 Hierarchical relations
Project activities are connected to the project by the BT relation or NT in the other direction. The hierarchical relationship used here is the hierarchical whole-part relation.

6.2.2.4 SECURITY

6.2.2.4.1 Concepts
SECURITY contains individual entities: risk and safety phrases (RS phrases). The lead terms are the Dutch RS phrases. RS phrases are translated in Danish (DEN), English (ENG), French (FRA), German (GER), Italian (ITA), Portuguese (POR), Spanish (SPA). The lead term is connected to the translated terms by a UF relation, renamed to the language abbreviation. Note that the cardinality constraint for the UF relation in SECURITY is 1:n.

6.2.2.4.2 Equivalence relations
RS phrases are translated in Danish (DEN), English (ENG), French (FRA), German (GER), Italian (ITA), Portuguese (POR), Spanish (SPA). The lead term is connected to the translated terms by a UF relation, renamed to the language abbreviation. Note that the cardinality constraint for the UF relation in SECURITY is 1:n.

6.2.2.4.3 Hierarchical relations
Not used.

6.2.2.5 TELEF.REGIST

6.2.2.5.1 Concepts
The concepts in TELEF.REGIST are individual entities. Lead terms are strings of the form "<register numbers>, <name>, <first name>" uniquely identifying individuals.

6.2.2.5.2 Equivalence relations
The lead term is connected to Initials, phone number, register number, name and user ID by a renamed UF relation. (AFK(1..1), TEL(1..n), REG(1..1), NAAM(1..n), UID)

6.2.2.5.3 Hierarchical relations
The lead term is connected to a department, location and state by a renamed BT relation (AFD, PLAATS, STATUS). (Cardinality =1:n)

6.3 Discussion
This chapter is part of the preparation step of the conversion method of ISO standards based thesauri to a Semantic Web representation as proposed in chapter 5. Also part of this step is the presentation of the conceptual model for standard thesauri as described by the ISO
standards for thesauri (ISO 2788 and ISO 5964). This is presented in section 3.2.1.2.

In this chapter, the Agfa thesauri have been analyzed to gain insight into their structure and semantics both the application semantics of the meta model as the real world semantics of these constructions. The thesauri are part of the BASIS document management system and managed by the Basis toolkit BASIS/TM (Thesaurus Manager). Therefore the BASIS/TM thesaurus data model has been presented first. Although using another terminology, this model can be mapped almost completely onto the ISO standard thesaurus model. The meta-models of the Agfa thesauri implemented in Basis are then described. These models make sometimes use of different terminologies for indicating relationships. These relationships can always be mapped onto the ISO standard relationships because they bear at least the same semantics. But often they have also more precise semantics. These semantics are not explicitly available in the underlying thesaurus model but implicitly intended by the thesaurus authors.

The technique used here to describe the structure and semantics of the thesauri meta models is by expressing them by means of natural language. This choice resulted from the fact that the semantics of the both the ISO standards and Basis thesaurus meta model terminology are described by natural language. The semantics of Agfa thesauri meta models implemented with the Basis system are even not described explicitly. Interviews with the thesaurus authors where needed to collect this information. This means that a completely described and machine interpretable syntax and semantics are not available. As a consequence, it is not possible to automatically translate the syntactical structures to another syntactical representation, in the case of this work RDF(S)/OWL XML without the risk of change in semantics.

A more formal description of the syntax and semantics could be made first and then translated to RDF(S)/OWL but then it is still necessary to have a clear understanding of the thesaurus models before making this formal specification. In fact by making the translation to RDF(S)/OWL of the thesauri meta models, a formal and machine processable description will result!

So, the goal of the descriptions in natural language is that this way the relations between the Agfa thesauri meta models, the Basis thesaurus meta model (implementing the Agfa thesauri) and the ISO standard thesauri meta model become clear. It is then possible to make a new thesaurus model which is able to represent not only the Agfa thesauri but also other standard based thesauri and to specify the semantics needed to deliver the desired thesaurus services as they are delivered now by the Basis system under the condition that the language used for this purpose has the possibilities to do this. As indicated already, the semantic web languages are probably suited to do this and even more. Because the RDF(S)/OWL constructions have clear an unambiguous model theoretic
semantics, it is possible to process and interpret the constructions automatically, using generic reasoners without the need to hard code the thesaurus services in the processing software.
7 Modeling thesauri using Semantic Web languages

7.1 Introduction

As a consequence of the conclusions of chapter 6, it is clear that just translating the meta-models of various existing thesauri to a RDF(S)/OWL representation will lead to as many semantic web thesaurus models as thesauri are existing. This will not only cause more work for carrying out the migration but also for the implementation of services for actually using the thesauri. Also interoperability of the various thesauri will be difficult because they have their own underlying data models.

The existence of a standard semantic web thesaurus model, capable of representing standard thesauri but also the non standard features could solve these problems.

In this chapter focus will be on the definition of a RDF(S)/OWL meta model for the representation of thesauri because RDF(S)/OWL has enough expressive power to completely describe the necessary semantics as needed by a generic reasoner to deliver the desired thesaurus services. Different approaches have been proposed by various research groups for modeling thesauri mostly using RDF(S). Also a few DAML+OIL and OWL proposals are available. Very recently a comparison has been made of the different approaches by . The different approaches will be described first and a thesaurus meta-model will be proposed.

7.2 Term-based thesaurus model

The main objects in a thesaurus are concepts and terms. (1986a) refers to it in the definition of a thesaurus as the vocabulary of a controlled indexing language, formally organized so that the a priori relationships between concepts are made explicit. Terms are representations of concepts.

In many thesauri, terms are the only type of objects modeled explicitly. But even if terms are the only thesaurus objects, concepts are often the underlying structures described by the terms.

In a term based thesaurus, terms are the only type of object explicitly considered. Inter term relations (BT, NT, RT, UF) are used to relate terms with other terms. The Agfa thesauri, discussed in section 6.2 are an example of term-based thesauri.

Different models of a term-based thesaurus with RDF and RDFS have been proposed.
A possibility is to model the concept Term as an RDFS class and the ISO relationships as properties. Special classes of terms, as Preferred Term and Top Term could be modeled as RDFS subclasses of the class Term. Adding other more specialized relationships is possible by making them sub properties of the standard properties. An example of a RDFS term based monolingual thesaurus model is the thesaurus of the Gateway to Education Materials (2004a). The thesauri are modeled as classes, relationships are modeled as rdfs:properties, represented in Figure 13 as Protégé slots.

Fig. 13: GEM rdfs properties

More specialized relationships, CR (Conceptually Related), FR (Functionally Related), PS (Physically Related) and TR (Temporally Related) are modeled as sub properties of the RT property.  

Fig. 14

In this approach, the actual thesaurus terms are considered as instances of the class Term and described by RDF statements.

Some thesauri are organizing terms into categories, additional to a hierarchy of terms. This could be modeled as a further extension of the term-based approach by the introduction of the concept Category explicitly as rdfs:class. All instances of Term are then linked to an instance of Category, using an additional property. This way, all thesaurus terms can be classified into categories. An example of this approach is the RDF Schema of the Ceres/NBII thesaurus (2003d). As mentioned by this approach allows the specific use of the instance_of relationship instead of using the more generic NT/BR relations between terms. But this could also be achieved by introducing the instance_of relation as a subproperty of the BT relationship which would be a simpler approach only requiring one additional property instead of an additional class and property.
Another approach is to model the terms themselves as RDFS classes. Then, terms can be classified using the rdfs:subClassOf property to model the BT and NT thesaurus relationships. The main problem with this approach is that the semantics of BT and NT and of rdfs:subClassOf are not similar. As already mentioned in section 3.2.1.2, the thesaurus hierarchical relationships are semantically overloaded and covers three logically different situations: the generic relationship, the hierarchical whole-part relationship and the instance relationship. By using the rdfs:subClassOf property to model this kind of relationships, this can lead to incorrect interpretation of RDF entailments. E.g. in a thesaurus, the parts of a car could be modeled by using the NT relation: car NT wheel. The same example in rdfs becomes:

```xml
<rdfs:Class rdf:ID="wheel">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="car"/>
  </rdfs:subClassOf>
</owl:Class>
```

Due to the rdfs semantics of subClassOf, one could conclude that a wheel is a car.

### 7.3 Concept-based thesaurus model

Terms are often language dependent representation of concepts, defined as abstract, universal psychical entities that serves to designate a category or class of entities, events or relations. In concept based thesaurus models, concepts are modeled explicitly as distinct classes. BT, NT, RT relationships could then be considered as inter concept relationships instead of inter term relationships. The other relationships, USE and USED FOR are concept/term relationships because they connect a concept with a language dependent representation of it. In recent work of SWAD Europe, Thesaurus activity, this approach has been chosen.

According to this approach could solve the problem of confusion when identical terms are used to indicate different concepts. An argument is that when two different concepts are pointing to the same term it is not clear if they are indeed different for the human reader. The distinction has to be made by using scope notes and full text explanations, or could be deduced by the position of the concept in the concept network.

Only, this argumentation seems to be also still valid in the term based approach. When two identical terms are created, for the system they are distinct (because it are two different objects and they exists in the same thesaurus). Disambiguation for the system is done by assigning a unique internal technical key to each term. In most systems it is done by assigning qualifiers to homonymous terms. This technique is also recommended by the ISO standards for thesauri en has also been used in the case of the Agfa thesauri. For the human reader, the distinction has to
be made further by explaining the term using scope notes. Or the meaning of the term could be deduced by its position in the graph. The solution to this problem when using the concept based approach, as proposed by, is the identification of concepts by a unique URI. But if this is a good solution then the same could be done for the term based approach: provide for each term a unique URI and use RDF:label to represent the term itself.

Another argument for the concept based approach from is that it captures the intuition that in practical thesaurus construction, the BT NT relationships reflects the extension of the concept, that is the resources which can be classified under those terms. They conclude that the broader/narrower relationships does after all represent a proper subclass inclusion, but not of the extensions of terms, but rather the extensions of the concepts. But this is still not the case when BT/NT relations are used as a part of relationship. And it is always possible in a term based approach to model that the relation between two terms are in fact relations between the extension of the concepts e.g. by using relations like BroaderConcept and NarrowerConcept.

The concept based approach introduces also some problems. The resulting meta model is more complicated. Also considering every thesaurus object as a concept conflicts with the fact that objects and concepts are not the same. The distinction between Concept and Object is due to the German philosopher Frege. According to Frege, any sentence that expresses a singular proposition consists of an expression (a proper name or a general term plus the definite article) that signifies an Object together with a predicate (the copula "is", plus a general term accompanied by the indefinite article or an adjective) that signifies (bedeuten) a Concept. Thus "Socrates is a philosopher" consists of "Socrates", which signifies the Object Socrates, and "is a philosopher", which signifies the Concept of being a philosopher.

In some cases the terms of a thesaurus are representing entities of one. E.g. the TELEF.REGIST thesaurus explained in section 6.2.2.5 contains concrete entities, names and telephone number. Considering these terms as concepts would not be (real world) semantically correct because it are individuals.

An example of a concept based RDF thesaurus model is proposed by (2000), the XML RDF Schema is given in appendix 11.4. Within the SWAD Europe Thesaurus Activity only a concept based RDF Schema for thesauri is proposed.

7.4 OWL and thesauri

The World Wide Web Consortium released on 10/02/2004 the Resource Description Framework (RDF) and the OWL Web Ontology Language (OWL) as W3C Recommendations.
In this section it will be investigated how the OWL specific features could be helpful in the representation of thesauri. Using OWL allows the definition of specific constraints on the data model. If the thesaurus data model is specified accurate enough, then it should be possible for a generic OWL reasoner to answer questions about a thesaurus as "give me the preferred term of term X" or "give me all the related terms of term Y". Also thesaurus management could be assisted by use of such tools: to check for instance that a preferred term has no use relation pointing to another preferred term. A complete thesaurus system could then be build using OWL and any generic OWL reasoner. At the moment there is no research known which proves the feasibility of this assumption. For an overview of OWL see also section 3.1.2.4.1.

7.4.1 OWL features
Some OWL features helpful for the representation of thesauri, are reviewed in this section.

7.4.1.1 OWL Light features

7.4.1.1.1 Cardinality
OWL cardinality restrictions are referred to as local restrictions, since they are stated on properties with respect to a particular class. That is, the restrictions constrain the cardinality of that property on instances of that class. If thesaurus objects are represented as classes, then they are not useful because not every thesaurus object has its own set of restrictions. If thesaurus objects refers to the same class e.g. to a class term or a class concept then the cardinality restrictions could be used to restrict certain properties (thesaurus relations).
It should be possible for instance to indicate that there is only one lead term. In the thesaurus example of section 6.2.2.5.2 different specialized equivalence relationships have also different cardinality constraints.
Cardinality expressions with values limited to 0 or 1 are part of OWL Lite. This permits to indicate 'at least one', 'no more than one', and 'exactly one'. Positive integer values other than 0 and 1 are permitted in OWL DL. owl:maxCardinality can be used to specify an upper bound. owl:minCardinality can be used to specify a lower bound. In combination, the two can be used to limit the property's cardinality to a numeric interval.

7.4.1.1.2 Equality - Inequality

7.4.1.1.2.1 Equivalent class
If thesaurus objects are modeled as classes themselves, then the feature 'equivalent class' could be used to indicate that they are equivalent, having the same instances. This way synonyms could be modeled.

7.4.1.1.2.2 SameAs
If thesaurus objects are modeled as individuals of classes, the feature SameAs could be used to indicate synonyms.

7.4.1.1.3 Property characteristics

7.4.1.1.3.1 InverseOf
One property may be stated to be the inverse of another property. If the property P1 is stated to be the inverse of the property P2, then if X is related to Y by the P2 property, then Y is related to X by the P1 property. This property could be used to indicate for instance that USE is the inverse of USE FOR or BT is the inverse of NT. If Term1 BT Term2, a reasoner could then deduce Term2 NT Term1.

7.4.1.1.3.2 TransitiveProperty
Properties may be stated to be transitive. If a property is transitive, then if the pair (x,y) is an instance of the transitive property P, and the pair (y,z) is an instance of P, then the pair (x,z) is also an instance of P.
E.g. the BT, NT thesaurus relations are transitive and this could be modeled using the TransitiveProperty. If it is stated that Term1 BT Term2 and Term2 BT Term 3 then a reasoner could deduce that Term 1 BT Term3.

7.4.1.1.3.3 SymmetricProperty
Properties may be stated to be symmetric. If a property is symmetric, then if the pair (x,y) is an instance of the symmetric property P, then the pair (y,x) is also an instance of P. E.g. the RT thesaurus relation could be defined as SymmetricProperty. If it is stated that Term1 RT Term2, a reasoner could deduce Term2 RT Term1.

7.4.1.1.3.4 FunctionalProperty
Properties may be stated to have a unique value. If a property is a FunctionalProperty, then it has no more than one value for each individual (it may have no values for an individual). This characteristic has been referred to as having a unique property. FunctionalProperty is shorthand for stating that the property's minimum cardinality is zero and its maximum cardinality is 1. E.g. the USE thesaurus relation could be modeled as a FunctionalProperty.

7.4.1.1.3.5 InverseFunctionalProperty
Properties may be stated to be inverse functional. If a property is inverse functional then the inverse of the property is functional. Thus the inverse of the property has at most one value for each individual. This characteristic has also been referred to as an unambiguous property.

7.4.1.1.4 Annotations
OWL Full does not put any constraints on annotations in an ontology. OWL DL allows annotations on classes, properties, individuals and ontology headers, but only under certain conditions.

Five annotation properties are predefined by OWL, namely:

- `owl:versionInfo`
- `rdfs:label`
- `rdfs:comment`
- `rdfs:seeAlso`
- `rdfs:isDefinedBy`

These annotations could be useful to describe the version of a thesaurus, of thesaurus terms or to annotate thesaurus objects. As specific annotation property could be defined to replace the Scope Note relation of a thesaurus.

7.4.1.1.5 Versioning
An `owl:versionInfo` statement generally has as its object a string giving information about this version, for example RCS/CVS keywords. Although this property is typically used to make statements about ontology’s, it may be applied to any OWL construct. It could be used for keeping track of changes in a thesaurus.

7.4.1.2 OWL DL features

7.4.1.2.1 disjointWith
Classes may be stated to be disjoint from each other. For example, Man and Woman can be stated to be disjoint classes. From this `disjointWith` statement, a reasoner can e.g. deduce an inconsistency when an individual is stated to be an instance of both. For thesaurus modeling this could be used to make explicit that the set of all preferred terms is disjoint with the set of non-preferred terms. A OWL reasoner could then automatically signal an inconsistency against this rule.

7.4.1.2.2 unionOf, complementOf, intersectionOf
OWL DL and OWL Full allow arbitrary Boolean combinations of classes and restrictions: `unionOf`, `complementOf`, and `intersectionOf`. For example, using `unionOf`, we can state that a class contains things that are either USCitizens or DutchCitizens. This features could be used to model that TopTerms never belong to the range of the NT relation.
7.5 Proposal for a generic OWL thesaurus model

According to the goals of this research as formulated in section 2.2, a generic OWL thesaurus model should be:

- capable of representing a thesaurus, built according to the ISO guidelines in a correct way, without changing the semantics of the existing model.
- it should be possible to automatically import existing thesaurus data, when build according to the ISO standards.
- it should also be possible to extend the model with other relationships as most thesauri have.
- the thesaurus data model should be precise enough to permit a generic reasoner to deliver the needed thesaurus services.

A concept based approach would introduce some problems as mentioned in section 7.3. and the real benefits in comparison with the term based approach are unclear. E.g. translating the Agfa TELEF.REG thesaurus to a concept based model would be semantically incorrect, because terms referring to objects would become concepts. This is avoided by considering the objects in a thesaurus just as terms, without making any assumption of what those terms represent. And this brings us to a term based approach, which has been chosen in this work in contrast with the SWAD Europe approach where only a concept based meta model for thesauri is proposed. Another argument for the term based approach is also that the resulting model is simpler and it is still possible to extend such a model in a way that it becomes a concept based model by connecting a term to a concept via an additional relation and a concept node. This can be done selectively for those terms representing concepts.

This proposal is based on the analysis of literature (see section 4.1), the analysis of both the ISO standards for thesauri (see section 3.2.1.2) and of the Agfa thesauri (see chapter 6).

7.5.1 Classes

The main object class of the thesaurus meta model is Term, representing terms of a thesaurus. Term instances are considered as proxies for objects or concepts (or whatever they represent) and are unique. Qualifiers are used as a disambiguation technique to prevent the existence of the same term with different meanings as proposed by (1986a;2003a). This approach has also been chosen for the Agfa thesauri.

Term has two subclasses: PreferredTerms and NonPreferredTerms. Both subclasses are disjoint: a term is either a preferred term or it is a non preferred term. The distinction between the two types of terms is important. E.g. for indexing, only preferred terms are used. The class of preferred terms has a subclass: the class of TopTerms, those terms which are on top of a term hierarchy. These classes and restrictions could be defined in OWL as follows:
7.5.1.1 Object Properties

The thesaurus relationships can be modeled as OWL object properties (or just as RDF properties, this would make no difference towards a thesaurus application).

Three classes of inter-term relationships are recognized in thesauri: the equivalence relationship, the hierarchical relationship and the associative relationship (see also section 3.2.1.2). They are very generic and it is always possible to relate a term to another by use of one of these three relationships because when it is not possible to use the equivalence or the hierarchical relationship the associative relationship could be used. They can be modeled as main RDF properties and the standard thesaurus relationships as sub properties according to the classification of Table 2. This sub property approach permits to add other relationships, with a more precise meaning. Providing them as a sub property of one of the tree main properties, permits to retrieve all terms related to a specific term by one of the three main properties by querying for the main property. An example is the “UD” relationship of the Agfa thesauri. It is an equivalence relationship with a more specific meaning then the USED-FOR relation. It is used to link a term with it’s German equivalent term. But it is often still needed to retrieve all equivalent terms of a specific term, independent of the actual property (UD or UF relation) that is used in an assertion about a term.

7.5.1.1.1 Associative relationships

A property AssociativeRelation has one sub property: the RelatedTerm property. The domain and range of this relationships are preferred terms. The RelatedTerm (RT) relationship is also symmetric. This leads to the following OWL definitions:
7.5.1.1.2 Equivalence relation

The EquivalenceRelation property has two sub properties: UsedFor and Use. Both sub properties are the inverse of each other. The domain of UsedFor are instances of the class PreferredTerm and the range instances of the class NonPreferredTerm. The domain of Use are instances of the class NonPreferredTerm and the range instances of the class PreferredTerm. In OWL this can be defined as follows:

```
<owl:ObjectProperty rdf:ID="EquivalenceRelation">
  <rdfs:range rdf:resource="#Term"/>
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relations between preferred and non-preferred terms.</rdfs:comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="Use">
  <rdfs:subPropertyOf rdf:resource="#EquivalenceRelation"/>
  <rdfs:range rdf:resource="#PreferredTerm"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#UsedFor"/>
  </owl:inverseOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="UsedFor">
  <rdfs:subPropertyOf rdf:resource="#EquivalenceRelation"/>
  <rdfs:domain rdf:resource="#NonPreferredTerm"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:about="#Use"/>
  </owl:inverseOf>
</owl:ObjectProperty>
```

7.5.1.1.3 Hierarchical relations

Hierarchical relations are modeled as a HierarchicalRelation property. BroaderTerm and NarrowerTerm are sub properties. The domain and range of HierarchicalRelation are instances of PreferredTerm. The range of the NarrowerTerm relation are instances of PreferredTerm without those instances which are also an instance of TopTerm. The property HierarchicalRelation is also defined as being transitive. BroaderTerm and NarrowerTerm are also the inverse property of each other.

In OWL this can be defined as follows:

```
<owl:ObjectProperty rdf:ID="HierarchicalRelation">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relations for use if equivalence or hierarchical relationships are not appropriate</rdfs:comment>
  <rdfs:domain rdf:resource="#PreferredTerm"/>
  <rdfs:range rdf:resource="#PreferredTerm"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2002/07/owl#SymmetricProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="BroaderTerm">
  <rdfs:range rdf:resource="#TopTerm"/>
  <rdfs:domain rdf:resource="#PreferredTerm"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="NarrowerTerm">
  <rdfs:range rdf:resource="#TopTerm"/>
  <rdfs:domain rdf:resource="#PreferredTerm"/>
</owl:ObjectProperty>
```

11/01/2012 Thesaurus representation with OWL 70 / 143
Note that the range restrictions for the relationships are needed to permit the automatic classification of a term as a preferred term or a non preferred term without having to state this explicitly. This is important for some thesaurus services e.g. to promote consistency in the assignment of indexing terms.

7.5.1.2 Data type properties

Data type properties could be used to relate instances of classes to instances of data types. E.g. to model thesaurus scope notes, the language of a term, versioning information …

The description in OWL:

```xml
<owl:DatatypeProperty rdf:ID="ScopeNote">
  <rdfs:domain rdf:resource="#PreferredTerm"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">This relation point to the scope note of a term</rdfs:comment>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="Language">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="DateAdded">
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
</owl:DatatypeProperty>
```

7.5.2 Non standard properties

The different Agfa thesauri contain various non-standard relationships.
They can now easily be added to the proposed model by making them sub-properties of the appropriate super-property. In general, the rdfs:subPropertyOf feature, which is transitive, permits to add other relations to the set of existing standard relations, while for retrieving it is still possible to get all the terms of a specific term related by a certain relation category independent of the specific actual property used.

Note that only for the thesaurus INVENT.DE, an OWL description will be provided because this thesaurus will also be translated completely as a test case in the context of this work. For the other Agfa thesauri it will be indicated how the non standard properties could be described in OWL, without providing them explicitly.

7.5.2.1 INVENT.DE

INVENT.DE has UD and ED as non-standard relationships. This relationship are used to link a term to a non-preferred German translation and could be modeled by making them sub-properties of the used-for and use properties.

A possible description in OWL:

```
<owl:ObjectProperty rdf:ID="ud">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relationship between preferred and non-preferred German term</rdfs:comment>
  <owl:inverseOf rdf:resource="#ED"/>
  <rdfs:subPropertyOf rdf:resource="#used-for"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="ed">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relationship between non-preferred German term and preferred term</rdfs:comment>
  <owl:inverseOf rdf:resource="#UD"/>
  <rdfs:subPropertyOf rdf:resource="#use"/>
</owl:ObjectProperty>
```

7.5.2.2 TELEF.REGIST

The TELEF.REGIST thesaurus contains several more specific used-for relations:
- AFK
- TEL
- REG
- NAAM

They can be defined as sub-properties of the UsedFor relation. Then it is still possible to retrieve from a certain term all it’s used-for terms, which will also deliver the term connected by the relations AFK, TEL, REG, NAAM.

Also some more specific BroaderTerm relations are defined:
- AFD
- PLAATS
- STATUS

They can be defined as sub-properties of the broader term relation. This way they can be used individually or collectively via BroaderTerm.
7.5.2.3 SECURITY
The SECURITY thesaurus contains specialized used-for relations:
- DEN
- ENG
- FRA
- GER
- ITA
- POR
- SPA
They are used to link a term to the equivalent term in another language. They could be defined as sub properties of the UsedFor property.

7.6 Discussion
A generic but extendible RDF(S)/OWL thesaurus model has been proposed. In this work, a term based approach has been chosen although previous work is often relying on a concept based approach.

Some benefits in favor of a concept based approach together with arguments that these problems are also solvable within a term based approach are:
- The concept based approach approach could solve the problem of confusion when identical terms are used to indicate different concepts. Also concepts have to be represented one way or another and that’s exactly what terms are for. Within a term based approach it is still possible to represent terms by unique ID’s and connect the terms as labels to the ID’s. There are also disambiguation techniques known to make terms with other meanings different e.g. within the ISO standards about thesauri.
- The concept based approach captures the intuition that in practical thesaurus construction, the BT NT relationships reflects the extension of the concept, that are the resources which can be classified under those terms.
  This should be no problem also for a term based thesaurus. One could for instance use the relationships BroaderConcept and NarrowerConcept which connects two terms to indicate that the concepts behind the terms are connected by the relation.

The concept based approach has also some disadvantages:
- The resulting meta model is more complicated.
- Also considering every thesaurus object as a concept conflicts with the fact that objects and concepts are not the same.
  In the Agfa thesaurus TEL.REGIST terms are representations of individuals and not concepts.
- A fundamental question with relation to the concept based approach is if it is possible to represent concepts at all. Concepts are related with meaning, connected to human thoughts. And we use ...terms to
communicate about these concepts! See also Figure 15 were the relations between symbols (Terms), thoughts and things are reflected (C.K. Ogden, 1923).

**Figure 15**

Because the concept based approach has no clear benefits and is not generic enough in a way that it is not always possible to translate a term based thesaurus into a concepts based model without introducing some semantic problems e.g. by considering an individual or real life object as a concept, the term based approach has been chosen (in contrast with SWAD Europe).

If explicit modeling of concepts is a necessity, it is always possible to extend such a term based model by defining a relation between a term and a concept node into a concept based model. But this is not necessary for an appropriate modeling of the Agfa thesauri and most other thesauri thinking about a term as a proxy for various kind of things like concepts but also real life objects.

Extendibility of the model can be achieved by defining the non standard relationships of the thesaurus meta model as sub properties of the main properties of the RDF(S)/OWL thesaurus model representing the associative, equivalence and hierarchical relationships. This seems to be always possible because the generic thesaurus relationships cover all possible relationships between terms. Other relationships can be considered as specializations of the generic relationships.

RDF(S)/OWL is capable of representing the semantics of a thesaurus meta-model in a way that a generic reasoner could provide the necessary thesaurus services. This will be investigated in the following chapters.
8 Thesaurus services

8.1 Introduction

A complete thesaurus system should be able to provide some basic services. Irrespective of their implementation, thesauri are often used in the same way, see also section 3.2. Therefore, it would be of advantage to have a standard service interface to a thesaurus system. In this chapter, generic thesaurus services will be described. RDF(S) querying or OWL reasoning could provide the necessary services supposing that the RDF(S)/OWL thesaurus meta model contains the necessary semantics. RDF(S)/OWL querying and reasoning will be explained in this chapter, some reasoners will be reviewed.

8.2 Generic services

SWAD Europe provides a list of 6 services. From this list, together with the information of section 3.2 about the use of thesauri and the requirements of the Agfa thesaurus managers following more extended list is proposed:

1. for a term X, give a list of equivalent terms (preferred, nonpreferred)
2. for a term X specify if it is a preferred or non preferred term
3. for a non preferred term X, give a list of all preferred terms
4. for a preferred term X, give a list of all non preferred terms
5. for a preferred term X, give his scope notes
6. for a term X, give all terms connected with a hierarchical or associative relationship (one level deep)
7. for a term X, give all the terms connected by relationship Y
8. for a term X give the complete graph (all assertions).
9. for a term X, give the transitive closure of the hierarchical terms
10. for a term X give all the top terms
11. for a thesaurus X, lookup all unrelated terms
12. for a thesaurus X, lookup all terms and relations coming into conflict with the thesaurus data model

8.3 Reasoning

The OWL DL language has a model-theoretic semantics such that very large fragments of the language can be directly expressed using so-called Description Logics. The design of OWL is based on the SHIQ family of Description Logics. This means that, with some restrictions, OWL documents can be automatically translated into T-boxes. The RDF-part of OWL documents can be translated to A-boxes (Volker Haarslev and Ralf Moller 2003b).
A reasoning system that can handle A-boxes and T-boxes could then be used to implement the thesaurus services as specified in section 8.2. See section 3.4 about Description Logics. In following sections some RDF(S)/OWL reasoners will be introduced.

8.3.1 OWL reasoners
W3C is providing a list of reasoners capable of reading OWL files. Two reasoner were used to test the OWL thesaurus model (T-box) and assertions (A-Box): Euler and Racer.

8.3.1.1 Racer
According to (Volker Haarslev and Ralf Moller 2003a), the RACER system is a knowledge representation system that implements a highly optimized tableau calculus for a very expressive description logic. It offers reasoning services for multiple TBoxes and for multiple ABoxes as well. The system implements the description logic SHIQ. RACER can also process OWL documents. Documents are interpreted with respect to the OWL DL languages. Racer is available as a standalone version, the Racer server. Clients can connect to RACER based on either TCP sockets or HTTP streams. For the test, the Racer Interactive Client Environment (RICE) was used.

8.3.1.2 Euler
Euler is an inference engine supporting logic based proofs of test cases. It is a backward-chaining reasoner enhanced with Euler path detection and will tell whether a given set of facts and rules supports a given conclusion using rule sets such as rdfs-rules and owl-rules (Jos De Roo 2004). It is implemented as open source programs Euler.java and Parser.java as well as C# program EulerSharp. Input format for Euler is N3, so the OWL/RDF files have to be converted e.g. by using Jena2. N3 is a language which is a compact and readable alternative to RDF’s XML syntax (Tim Berners-Lee 2001b). Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, including a rule-based inference engine (2004b).

8.4 RDF/RDFS querying
Instead of using a OWL aware reasoner, it is also possible to query the thesaurus with a RDF(S) aware query handler. Of course the OWL specific features of the thesaurus data model will be not understood by these kind of query handlers. This means that a RDF(S) query handler is not aware of the inverse, transitive and symmetric properties defined in the OWL data model. As a consequence, if statements are made about terms using an inverse property (e.g. termA broader-term termB) always the inverse of this statement must be added explicitly (termB broader-term termA). The same counts for a symmetric property, e.g. if a statement exist “termA related-term termB” then also a statement has to be provided “termB related-term termA”. For the Agfa thesauri this is not a problem because the output from Basis+ is complete with respect to this issue.
Also the transitive closure can not be calculated by a RDFS query handler because it is not aware of the transitivity of certain properties, e.g. the broader-term and narrower term properties. But this could be easily handled by iteratively querying the database with the query output of a previous query as input for the next query.

In general, when a certain semantic construction is not possible within the thesaurus model specification language, the semantics have to be hard coded into the application using the thesaurus model and thesaurus data to deliver the desired thesaurus services.

In this work, Sesame en CWM will be used to query the thesaurus.

8.4.1 Sesame

Sesame is a system for the storage an querying of RDF and RDFS information. Sesame allows persistent storage of RDF data and schema information, and provides access methods to that information through export and querying modules.

Part of Sesame is SeRQL ("Sesame RDF Query Language", pronounced "circle") a RDF/RDFS query language. It combines features of other (query) languages (RQL, RDQL, N-Triples, N3) and adds some of its own.

Some of SeRQL's most important features are (2004d):

- Graph transformation.
- RDF Schema support.
- XML Schema data type support.
- Expressive path expression syntax.
- Optional path matching.

For persistent storage of RDF data, Sesame can use a relation database e.g. Oracle, MySQL or PostgreSQL. Sesame itself is kept database independent, all DBMS specific code is concentrated in a single architectural layer of Sesame, the Storage and Inference Layer (SAIL). Sesame’s functional modules, e.g. the RQL query engine, are clients of the SAIL API. In Sesame, RQL queries are translated into a set of calls to the SAIL, the main bulk of the actual evaluation of the RQL query is done in the RQL engine itself.

For testing the thesaurus, Sesame is used with a MySQL database for storage of the RDF(S) data.

8.4.2 CWM

CWM is a Semantic Web program that can do the following tasks:

- Parse and pretty-print the following RDF formats: XML, RDF, Notation3, and NTriples
- Store triples in a queryable triples database
- Perform inferences as a forward chaining FOPL inference engine
- Perform built-in functions such as comparing strings, retrieving resources, all using an extensible built-ins suite

CWM was written in Python from 2000-10 onwards by Tim Berners-Lee and Dan Connolly of the W3C (Tim Berners-Lee 2001a).

8.5 Discussion and Conclusion

Irrespective of their implementation, thesauri are often used in the same way. The most common thesaurus services are described in this chapter. A practical thesaurus system should be able to provide these services. In theory, a generic OWL reasoners could be able to provide them as it is shown in section 7.5 that it is possible to construct a thesaurus OWL meta-model with a complete representation of the model semantics of the thesaurus constructions. An OWL reasoner should then be able to deliver the services in a correct way by making use of the model theoretic semantics of the thesaurus meta-model constructions during the internal reasoning process.

Also RDF(S) query systems could do this, at least partly, because a significant part of the thesaurus meta-model presented in section 7.5 can be expressed with only RDF(S) constructions. Only the inverse, transitive and symmetric properties defined in the thesaurus meta-model can not be described with RDF(S) and therefore also not interpreted by an RDF(S) reasoner. In this case, in order to be able to deliver complete thesaurus services, the thesaurus data should contain two RDF triples for each relation used with symmetric and inverse properties, to represent these properties implicitly in the dataset. This is in the case of the conversion of the Agfa thesauri not a problem as the Basis thesaurus data are complete with respect to this issue.

To calculate the transitive closure when using a RDF(S) reasoner, for the “BroaderTerm” and “NarrowerTerm” constructions in a thesaurus, the RDF triples database could be iteratively queried with the query output of a previous query as input for the next query. This way, the transitive property of the “BroaderTerm” and “NarrowerTerm” relations are hard coded into the thesaurus service application instead of explicitly modeled in the thesaurus meta-model.

The motivation that also RDF(S) reasoner are tested is that it can be expected that the calculation complexity in the case of OWL reasoners is much higher than for RDF(S) reasoners as it is shown in literature that adding more expressive power to a Description Logic can result in much higher calculation complexity in terms of space and time needed to carry out the reasoning, see also section 3.4 about Description Logics.

In this work, two OWL reasoners are tested, Euler and Racer, described in this chapter. No references in literature have been found to comparable tests for the implementation of a thesaurus system. Also two RDF(S)
reasoner, CWM and Sesame are described in this chapter and are tested within this work. The test of the reasoners is the subject of chapter 8.5.
9 Test of the RDF(S)/OWL thesaurus

9.1 Introduction

In chapter 7, a standard extendible RDF(S)/OWL thesaurus meta model has been proposed. In chapter 9, the minimal necessary thesaurus services have been defined which could be implemented by using generic RDF(S)/OWL query handlers and reasoners. Two OWL reasoners and two RDF(S) reasoners are proposed.

In this chapter it will be investigated if a real life thesaurus system effectively could be implemented based on the standard thesaurus meta-model and whether OWL or RDF(S) reasoners could provide the desired services.

To test out the semantic web thesaurus system, a knowledge base is made with the DE thesaurus converted to a set of RDF assertions, representing the A-box and the proposed generic OWL thesaurus model converted to the T-Box. This knowledge base was then tested with Racer, Euler, Sesame and CWM. See also chapter 8 for a short description of the reasoners.

The results of the tests are described in following sub-sections.

9.2 Test set-up

A set of queries has been defined, covering the thesaurus services as described in chapter 8. The queries are presented in Table 4:

<table>
<thead>
<tr>
<th>ID</th>
<th>Description of the test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>for term X, find all terms related by equivalence relation</td>
</tr>
<tr>
<td>2</td>
<td>for term X, find all terms related by narrower-term</td>
</tr>
<tr>
<td>3</td>
<td>for term X, find all terms related by broader-term</td>
</tr>
<tr>
<td>4</td>
<td>for term X, find all scope notes</td>
</tr>
<tr>
<td>5</td>
<td>for term X, find all assertions</td>
</tr>
<tr>
<td>6</td>
<td>for a preferred term X, indicate term type</td>
</tr>
<tr>
<td>7</td>
<td>for a non preferred term X, indicate term type</td>
</tr>
<tr>
<td>8</td>
<td>find all preferred terms</td>
</tr>
<tr>
<td>9</td>
<td>find all non preferred terms</td>
</tr>
<tr>
<td>10</td>
<td>for term X, find all terms related by hierarchical-relation</td>
</tr>
<tr>
<td>11</td>
<td>Find all top terms</td>
</tr>
</tbody>
</table>

Table 4: set of queries

The queries were translated to the system query syntax and carried out first on a limited subset of the complete DE thesaurus. This way the results can be checked very easy. The limited test set of assertions about terms in RDF format is appended in section 11.6.
In a second test, the complete DE thesaurus containing about 137000 terms was converted to RDF, loaded into the systems under investigation and queried.

### 9.3 Test with Racer
Racer (Volker Haarslev and Ralf Moller 2003b) was loaded with the limited test set of term assertions. RICE was used as a Racer client. The queries of Table 4: set of queries were translated to the Racer query syntax. The queries and test results are available in appendix 11.7.

The tests were mostly successful. Racer is capable of automatically classifying the terms as preferred or non-preferred term. Finding out which terms are also top terms did not succeed. Also the transitivity of the hierarchical-relation is not inherited by the narrow-term and broader-term relations. When asked for all the terms related by a hierarchical relation to the term ISO, a correct answer was provided although no explicit assertions were provided for the term ISO. This shows that the OWL model and a generic OWL reasoner are capable to provide the thesaurus services and to make implicit information explicit (except for the transitive closure).

A second test was done to check if Racer was capable to handle a realistic amount of thesaurus data: the complete DE thesaurus, containing 137077 terms was converted to OWL and loaded in Racer. This process took 8 minutes. Then asking a simple questions, looking for the narrower-terms of a given term, took about 20 minutes of processor time (100%) on a Pentium M (Dell Latitude D600) and consumed 300 MB of RAM and ended with a crash of Racer. This may indicate that Racer is not capable of handling large amounts of data because the computation complexity is too high, although further investigations are needed to exactly determine the cause of the system overload (what is not in the scope of this work).

### 9.4 Test with Euler
The same tests were done using Euler. See appendix 11.8 for detailed results. Euler showed the same results as Racer but also the same problems: not able to classify terms as top terms, not calculation of the transitive closure, not able to handle the total DE thesaurus.

### 9.5 Test with Sesame
Small modifications were needed to use the OWL thesaurus data model with Sesame: owl:ObjectProperty was replaced by rdf:Property, no other changes were made. The changed thesaurus data model for Sesame is presented in appendix 11.5. Sesame was first tested with a small part of the thesaurus, test set 1, for queries and results see appendix 11.8.

The most important conclusions are that:

- Sesame can infer the type of a term, a preferred term or a non preferred term because domain and range restrictions are also RDFS features.
Sesame can also infer all assertions made about the properties hierarchical-relation, equivalence-relation and associative-relation because the actual used properties are declared as sub-properties. And SubpropertyOf is an RDFS feature.

Because InverseOf is not a RDFS feature, the way queries are build is important with a term as subject or object. E.g. if from a term only the statement “termA broader-term termB” and not “termB narrower-term termA”, then a query looking for “termB hierarchical-relation termX?” will return nothing. For the RDF/XML representation of the Agfa thesauri, this is no issue because from a certain statement the inverse statement is always explicitly present (if applicable depending on the type of property).

TransitiveProperty is an OWL feature, as a consequence it is not possible to extract all hierarchical related terms of a specific term at once, only terms directly related (one level deep) are returned. This could be solved by iteratively expanding the query with the terms returned.

Secondly, the whole DE thesaurus was uploaded to the Sesame server. This process took about 10 minutes on a Pentium M 1.6 MHz computer with 512 MB ram. Query results were delivered fast, for most cases less then 80 ms (indicated by Sesame) are needed. This permits to use Sesame in a production environment.

9.6 Tests with CWM

CWM was also first tested with a small part of the thesaurus. See section 11.9 for a detailed description of the tests and the test results. The CWM results were comparable with the Sesame results as CWM is only RDF(S) aware.

9.7 Discussion and conclusion

The OWL reasoners Racer and Euler in combination with the proposed RDFS/OWL thesaurus meta model and the RDF thesaurus data are able to provide most of the desired thesaurus services. OWL is providing the possibility to define additional properties to the thesaurus relations (named properties in RDF(S)/OWL) such as reflexive, transitive and symmetric properties. This permits the reasoners to infer new facts about the thesaurus data not available explicitly in the thesaurus, based on thesaurus meta model constructions and the well defined model theoretic semantics of the properties. Two problems were detected: the inability to derive TopTerms based on the description in the meta model and the automatic calculation of the transitive closure to detect all narrower and broader terms of a given input term. Further research is needed to investigate the reasons as this is beyond the scope of this work. When tested with a realistic amount of data, both reasoners were unable to deliver the requested services. Probably this is due to the to high computation complexity in terms of space and time for this amount of data.
Also here is further research needed to pinpoint the problem exactly and to propose solutions for it as this is also beyond the scope of this work.

The rationale of having a complete description of the thesaurus model in OWL is that this can be of great help for thesaurus management, that the amount of thesaurus data could be reduced significantly because symmetric and inverse terms could be derived automatically and no ‘knowledge’ of the thesaurus model must be hard coded in the application interfacing to the OWL reasoner.

It has been demonstrated that it is also possible to make use of an RDF(S) query language. An RDF query language is not aware about the OWL semantics and as a consequence assertions making use of symmetric and inverse properties must be made in both directions (e.g. the USE and UF relation) or this must be handled by the application interfacing to the RDF query engine as is also the case for the transitive relations.

When querying was done on a complete thesaurus of about 117000 terms, only Sesame, a generic system for storing and querying RDF, was performing well. This is the only system of the test which probably is useful in a production environment although this was not tested out as this was beyond the scope of this work.
10 Conclusions and future work

In the course of this work, it has been demonstrated that it is possible to represent an extensible thesaurus meta-model with OWL. This model contains the necessary meta-model semantics to enable generic OWL reasoners to deliver the desired thesaurus services. This way it is possible to separate the semantics of in this case the thesaurus application and the program logic contained within the reasoners. This results in a flexible and easy to build and maintain thesaurus system. When changes are needed, e.g. to implement additional non standard thesaurus behavior, only the OWL meta-model has to be extended or adapted.

With respect to the tested reasoners, Racer and Euler, some problems were indicated as explained in chapter 9. Both Racer and Euler could not derive the top terms of a term tree, based on the model semantics contained by the OWL thesaurus meta-model and the thesaurus data. The biggest problem was that both reasoners were not capable to cope with a thesaurus of a realistic size. Probably, the computation complexity was too high. Further research is needed to pin-point the exact causes and to indicate some possible solutions. Also the constructions used within the thesaurus meta-model can influence the computational behavior. Therefore it would make sense to compare different OWL meta-models with respect to the computational complexity. At the moment, the best solution is to rely on a simpler RDF(S) based thesaurus meta-model. The consequence of this is that symmetric and inverse terms could not be derived automatically based on RDF assertions in one direction. As a result, the thesaurus data must be complete. Another consequence is that some ‘knowledge’ of the thesaurus model must be hard coded in the application interfacing to the RDF(S) reasoner, e.g. to derive the transitive closure of a term involved in broader and/or narrower relations. But both tested RDF(S) reasoners, CWM and Sesame, could deliver the desired thesaurus services also when a complete thesaurus of about 150,000 terms was loaded. Based on the results of this work, it can be expected that they are able to deliver the desired thesaurus services under production conditions although additional tests are needed to confirm this.

The in chapter 7 proposed and implemented thesaurus meta-model is a so called terms based thesaurus model. This in contrast with the SKOS core meta model of W3C. The rationale for this is that the concept based approach has no clear benefits and even can introduce some new problems converting an existing thesaurus. In this work, it has been argued that the problems which could be resolved by using a concept based approach are also resolvable within a term based approach. The concept based thesaurus model is not generic enough in a way that it is not always possible to translate an existing thesaurus into a concepts based model without introducing some semantic problems e.g. by considering an individual or real life object as a concept. This is related
with the fact that the terms of a thesaurus can point to anything: besides
that they can serve as proxies for concepts they are often names and ID’s
indicating individuals and objects.

The extensibility of the proposed OWL thesaurus meta-model has been
demonstrated for the case of the Agfa thesauri where non standard
relationships were added to the standard thesaurus meta-model by
defining the non standard relationships as sub properties of the main
properties of the RDF(S)/OWL thesaurus model, representing the
associative, equivalence and hierarchical relationships. This seems to be
always possible as the generic thesaurus relationships cover all possible
relationships between terms. Other relationships can be considered as
specializations of the generic relationships. This assumption has not been
investigated extensively and could be a subject for future work. Also the
introduction of multilingualism in the meta-model has not been
investigated extensively as this was available in the Agfa thesaurus model
by connecting a term to a translation in another language by using a
specialized ‘used for’ relation. Concerning the extensibility of the model it
should be noticed that the use of OWL offers more expressive power
compared to RDF(S) but that a possible practical limit is the increased
calculation complexity leading to long response times for and extended set
of thesaurus data.

The migration method proposed in this work is based on previous work of
but has been adapted as argued in section 5.6. The point of
departure of the in this work proposed method is the existence of a
generic RDF(S)/OWL thesaurus meta-model. The benefits are obvious: a
generic meta-model can be reused in most cases when thesauri are
implemented based on semantic web technology. Working from the start of
to a known goal seems to be more purposeful and efficiently. When such a
model exists, also the meta-model semantics are described already
formally, based on the model theoretic semantics of RDF(S)/OWL.
Therefore the proposed method starts with an analysis of the existing
thesaurus to identify the source thesaurus meta-model semantics to be
able to define a mapping between the source meta model and the
destination meta model in a semantics preserving way. In most cases this
is not automatically possible as the source meta model semantics are not
formally described and even not explicitly available. Therefore this is also
the most critical step as it is possible to interpret incorrectly the source
meta model semantics. The mapping between both meta-models is called
a semantic mapping because the intended semantics of the source meta-
model must be preserved. During this mapping activity it becomes clear if
the generic OWL/RDF(S) meta model needs to be extended in order to be
able to represent all needed relations between terms. The two following
steps are then pure syntactic conversions of the thesaurus data to the new
data structure using the terminology of the RDF(S)/OWL thesaurus meta
model. The division in two steps is suggested for practical reasons: the
first syntactical step is as simple as possible translation (from the
viewpoint of the implementation) to a structure preserving XML syntax of the source thesaurus data. The second step translates the XML source data into RDF assertions by using XSLT declarations and an XSLT processor as this is a convenient and standard way to describe and carry out the syntactic conversion between different XML formats. In fact with XSLT it is possible to separate the rules for conversion from the actual conversion logic yielding an easier to manage conversion process.
11 Appendix

11.1 Python source code for conversion of the thesaurus from Basis text format to a simple XML representation.

# Bert De Winter - 3866
# RDM/Technical Imaging
# 14/05/2004

# compiler for converting a Basis Thesaurus to OWL
# import LL2XML

# parser object
class Parser:
    # parser makes for every thesaurus block of the Basis thesaurus text dump (a term with relations (BT, NT, USE, SN, ...and related terms)
    # a list of lists. A block contains a term with his related terms (all kind of relations). A block end with an empty line.
    # The first list contains the main term with tag <term>
    # the second en following lists contains a relation tag and a term or string
    # example:
    # [['<term>', 'ABLATION'], ['UF', '10006'], ['UF', 'ABLATIE'], ['BT', 'REMOVAL'], ['RT', 'PIT']]

    # relation tag
tag = """"
    # list of lists
termlist = []

def __init__(self, file):
    self.xml_out = file

def escape(self, s):
    # """ """Replace special characters '&', '"', '<', '>' and '""" by XML entities."""
    s = s.replace("&", "&amp;")  # Must be done first!
    s = s.replace("\", "\")
    s = s.replace("<", "&lt;")
    s = s.replace(">", "&gt;")
    s = s.replace("", "&quot;")
def WriteXML(self, list):
    if list == []:
        return
    string = ""
    x = 0
    tag = ""
    for terms in list:
        x = x + 1
        if (x == 1):
            if (terms[0] == ":"):
                # if new block not contains a term, skip
                return
            try:
                tag = terms[0].lower()
                string = string + "<" + tag + "+" + self.escape(terms[1]) + "+" + "<" +
                tag + "+" + self.escape(terms[1]) + "+" + "+" + "" +
                tag + "+" + self.escape(terms[1]) + "+" + "" +
                except ValueError:
                    pass
            else:
                try:
                    rtag = terms[0].lower()
                    string = string + "<" + rtag + "+" +
                    self.escape(terms[1]) + "+" + rtag + "+" +
                    except ValueError:
                        pass
                string = string + "+" + tag + "+"
                self.xml_out.write(string)

def Parse(self, line):
    items = []
    items = line.split( ' ')
    stripItems = [""
    for item in items:
        stripItems.append(item.strip())
        if len(stripItems) == 2 and stripItems[1] == ":":
            # new/end of block detected
            self.tag = ""
            print self.termlist
            self.WriteXML(self.termlist)
        elif len(stripItems) == 2 and stripItems[1] != ":":
            # new block detected, tag is <term>
            self.tag = "term"
            pair = list()
            pair.append(self.tag)
            elem = stripItems[1]
            pair.append(elem)
self.termlist = [pair]
print self.termlist

elif len(stripItems) == 4:
    # new tag detected
    self.tag = stripItems[2]
    pair = []
    pair.append(self.tag)
    pair.append(stripItems[3])
    self.termlist.append(pair)

elif len(stripItems) == 6:
    if self.tag == 'SN':
        pair = self.termlist[-1]
        str1 = pair[0]
        str2 = pair[1]
        str2 = str2 + ' ' + stripItems[5]
        pair = [str1, str2]
        del self.termlist[-1]
        self.termlist.append(pair)
    else:
        pair = [self.tag, stripItems[5]]
        self.termlist.append(pair)

### Main program
### open file

file_in = open(r"G:\Rdm_Cs\In\Business Process Groups\General ICS_IBT support\Semantic web\owl\Copy of DE.put", "r")
xml_out = open(r"G:\Rdm_Cs\In\Business Process Groups\General ICS_IBT support\Semantic web\owl\DE_cor.xml", "w")
xml_out.write (r'<?xml version="1.0" encoding="iso-8859-1"?>')
xml_out.write("<de>")
myParser = Parser(xml_out)
for line in file_in.readlines():
    # print line
    myParser.Parse(line)

file_in.close
xml_out.write("</de>")
xml_out.close

11.2 The GEM RDF Schema thesaurus model

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE rdf:RDF (View Source for full doctype...)>
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <!-- Description of this Schema -->
  <rdf:Description rdf:about="http://purl.org/gem/NISO-Z3919/"
    <dc:title>Monolingual Thesauri Vocabulary</dc:title>
    <dc:publisher>The GEM Consortium</dc:publisher>
    <dc:description>Describes the relationships among thesauri terms. Uses a structure based on the NISO Z39.19 standard.</dc:description>
    <dc:language>English</dc:language>
    <dc:date>2001-04-11</dc:date>
  </rdf:Description>
- <rdfs:Class rdf:about="http://purl.org/gem/NISO-Z3919/NISO-Z3919">
  <rdfs:label>Monolingual Thesauri</rdfs:label>
  <rdfs:comment>Provides a schema for Monolingual Thesauri, based on NISO Z39.19-1993. Instances are required to contain a rdfs:label and a rdf:value.</rdfs:comment>
</rdfs:Class>
  <rdfs:label>Broader Term</rdfs:label>
  <rdfs:comment>A descriptor to which another descriptor or multiple descriptors are subordinate in a hierarchy. The relationship indicator for this type of term in BT.</rdfs:comment>
</rdf:Property>
  <rdfs:label>Related Term</rdfs:label>
  <rdfs:comment>A descriptor that is associatively but not hierarchically linked to another description in a thesaurus. The relationship indicator for this type of descriptor is RT.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="http://gemstar.ischool.washington.edu/2.0/gem#catal going"/>
</rdf:Property>
- <rdf:Property rdf:about="http://purl.org/gem/NISO-Z3919/NT">
  <rdfs:label>Narrower Term</rdfs:label>
  <rdfs:comment>A descriptor that is subordinate to another descriptor or to multiple descriptors in a hierarchy. The relationship indicator for this type or term is NT.</rdfs:comment>
</rdf:Property>
11.3 The CERES/NBII Thesaurus RDF Schema

```xml
<rdfs:isDefinedBy
</rdf:Property>

<rdfs:Property rdf:about="http://purl.org/gem/NISO-Z3919/USE">
    <rdfs:label>Use</rdfs:label>
    <rdfs:comment>Leads from a nonpreferred term to the descriptor.</rdfs:comment>
</rdf:Property>

<rdfs:isDefinedBy
</rdf:Property>

<rdfs:Property rdf:about="http://purl.org/gem/NISO-Z3919/UF">
    <rdfs:label>Top Term</rdfs:label>
    <rdfs:comment>The broadest descriptor in a thesaurus hierarchy, sometimes indicated by the abbreviation TT.</rdfs:comment>
</rdf:Property>

<rdfs:isDefinedBy
</rdf:Property>

<rdfs:Property rdf:about="http://purl.org/gem/NISO-Z3919/HN">
    <rdfs:label>History Note</rdfs:label>
    <rdfs:comment>A note in a term record in a thesaurus that provides the date of entry of a descriptor as well as the history of modification to its scope, relationships, etc.</rdfs:comment>
</rdf:Property>

<rdfs:isDefinedBy
</rdf:Property>

<rdfs:Property rdf:about="http://purl.org/gem/NISO-Z3919/SCOPE">
    <rdfs:label>Scope Note</rdfs:label>
    <rdfs:comment>A note following a descriptor explaining its coverage, specialized usage, or rules for assigning it.</rdfs:comment>
</rdf:Property>

<rdfs:isDefinedBy
</rdf:Property>
</rdf:RDF>

11/01/2012

Thesaurus representation with OWL
Thesaurus representation with OWL
11.4 RDF/XML Thesaurus Schema

<rdf:RDF xml:lang="en"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
  <rdfs:Class rdf:ID="Concept">
    <rdfs:comment>
      A unique concept defined within a thesaurus. Instances use the rdfs:isDefinedBy property with a vocabulary namespace as its value, to indicate the vocabulary to which the concept belongs.
    </rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:ID="Term">
    <rdfs:comment>
      Instances of this class represent the written forms of Concepts. The string is given by the rdf:value of Term.
    </rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:ID="ScopeNote">
    <rdfs:comment>
      The value of this optional resource is a scope note: a note attached to a term to indicate its meaning within an indexing language
    </rdfs:comment>
  </rdfs:Class>
</rdf:RDF>
<rdfs:Class rdf:ID="TermUsageValue">
  <rdfs:comment>
The value of the property: termUsage. It can take one of two values: 'preferred' or 'nonPreferred'.
  </rdfs:comment>
</rdfs:Class>

<rdf:Property ID="broaderConcept">
  <rdfs:comment>
  This schema does not define a property 'narrowerConcept', but applications can assume the existence of a property narrowerConcept such that if:
  {broaderConcept,ConceptA,ConceptB}, then {narrowerConcept,ConceptB,ConceptA} is true.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Concept"/>
  <rdfs:range rdf:resource="#Concept"/>
</rdf:Property>

<rdf:Property ID="relatedConcept">
  <rdfs:comment>
The relatedConcept is commutative, such that if:
  {relatedConcept,ConceptA,ConceptB}, then {relatedConcept,ConceptB,ConceptA} is true.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Concept"/>
  <rdfs:range rdf:resource="#Concept"/>
</rdf:Property>

<rdf:Property ID="indicator">
  <rdfs:comment>
  A mandatory property of a Concept whose value is the Term instance representing a written form of the Concept. A Concept may have as an indicator more than one Term. A Term may only be an indicator of one Concept.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Concept"/>
  <rdfs:range rdf:resource="#Term"/>
</rdf:Property>

<rdf:Property ID="conceptCode">
  <rdfs:comment>
  An optional property for any code assigned to the thesaurus concepts.
  </rdfs:comment>
</rdf:Property>
<rdfs:domain rdf:resource="#Concept"/>
</rdf:Property>

<rdf:Property ID="scope">
  <rdfs:comment>
    This optional property has as its value an instance of 
    the resource ScopeNote.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Concept"/>
  <rdfs:range rdf:resource="#ScopeNote"/>
</rdf:Property>

<rdf:Property ID="lang">
  <rdfs:comment>
    Optional property that can be used to give the language 
    of a Term instance. The codes from "ISO 639:1988, 
    Code for the representation of names of languages" should 
    be used as the values for this property.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Term"/>
</rdf:Property>

<rdf:Property ID="termUsage">
  <rdfs:comment>
    This optional property indicates whether the Term 
    instance is the 'preferred or 'nonPreferred' textual 
    expression of the Concept instance that is 'indicated' 
    by the Term, for a given language.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#Term"/>
  <rdfs:range rdf:resource="#TermUsageValue"/>
</rdf:Property>

<rdf:Description rdf:ID="preferred">
  <rdf:type rdf:resource="#TermUsageValue"/>
</rdf:Description>

<rdf:Description rdf:ID="nonPreferred">
  <rdf:type rdf:resource="#TermUsageValue"/>
</rdf:Description>

</rdf:RDF>
11.5 RDF/XML thesaurus model for Sesame

```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns="http://www.agfa.com/w3c/2004/05OWLThesaurus#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xml:base="http://www.agfa.com/w3c/2004/05OWLThesaurus#"
xmlns:oth="http://www.agfa.com/w3c/2004/05OWLThesaurus#">
  <owl:Ontology rdf:about="OWLThesaurus"/>
  <owl:Class rdf:about="#PreferredTerm">
    <rdfs:comment
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The class of all preffered thesaurus terms</rdfs:comment>
    <owl:disjointWith>
      <owl:Class rdf:about="#NonPreferredTerm"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Term"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Term">
    <rdfs:comment
      rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The class of all thesaurus terms. Every term is considered as unique altough the labels could be the same. A term can be considered as a proxy for a concept or object.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:about="#NonPreferredTerm">
    <owl:disjointWith rdf:resource="#PreferredTerm"/>
    <rdfs:subClassOf rdf:resource="#Term"/>
  </owl:Class>
  <owl:Class rdf:about="#TopTerm">
    <rdfs:subClassOf rdf:resource="#PreferredTerm"/>
  </owl:Class>
  <owl:Class rdf:about="#Label">
    <rdfs:comment>
      <rdf:Property rdf:about="#has-label">
        <rdfs:range rdf:resource="#Label"/>
      </rdf:Property>
    </rdfs:comment>
  </owl:Class>
</rdf:RDF>
```
<rdfs:domain rdf:resource="#Term"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">points to the label of a term</rdfs:comment>
<owl:inverseOf>
  <rdf:Property rdf:about="#label-of"/>
  <owl:inverseOf/>
</rdf:Property>
</rdf:Property>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">this relationship could be used to indicate any kind of relation between terms other than equivalence relations or hierarchical relations.</rdfs:comment>
</rdf:Property>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relations between preferred and non-preferred terms.</rdfs:comment>
<rdfs:range rdf:resource="#PreferredTerm"/>
<rdfs:domain rdf:resource="#Term"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relationship between non-preferred and preferred term</rdfs:comment>
</rdf:Property>
<owl:inverseOf rdf:resource="#used-for"/>
</rdf:inverseOf>
</rdf:Property>
</rdf:Property>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relationships between preferred and non-preferred term</rdfs:comment>
<owl:inverseOf rdf:resource="#use"/>
</rdf:InverseOf>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdfs:domain rdf:resource="#Term"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">their relationship could be used to indicate any kind of relation between terms other than equivalence relations or hierarchical relations.</rdfs:comment>
</rdf:Property>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relationships between preferred and non-preferred terms.</rdfs:comment>
<rdfs:range rdf:resource="#PreferredTerm"/>
<rdfs:domain rdf:resource="#Term"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relationship between non-preferred and preferred term</rdfs:comment>
</rdf:Property>
<owl:inverseOf rdf:resource="#used-for"/>
</rdf:InverseOf>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
</rdf:Property>
<rdfs:subPropertyOf rdf:resource="#equivalence-relation"/>
</rdf:Property>

<brdf:Property rdf:about="#associative-relation">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relations for use if equivalence or hierarchical relationships are not appropriate</rdfs:comment>
<rdfs:domain rdf:resource="#PreferredTerm"/>
<rdfs:range rdf:resource="#PreferredTerm"/>
</rdf:Property>

<brdf:Property rdf:about="#broader-term">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">relation from a subordinate term to a superordinate term</rdfs:comment>
<rdfs:domain rdf:resource="#PreferredTerm"/>
<rdfs:range rdf:resource="#PreferredTerm"/>
<owl:inverseOf><rdf:Property rdf:about="#narrower-term"></owl:inverseOf>
</rdf:Property>

<brdf:Property rdf:about="#narrower-term">
<rdfs:subPropertyOf rdf:resource="#hierarchical-relation"/>
</rdf:subPropertyOf>
<rdfs:range rdf:resource="#PreferredTerm"/>
<owl:inverseOf><rdf:Property rdf:about="#broader-term"></owl:inverseOf>
</rdf:Property>

<brdf:Property rdf:about="#broader-term">
<rdfs:subPropertyOf rdf:resource="#hierarchical-relation"/>
</rdf:subPropertyOf>
<rdfs:range rdf:resource="#PreferredTerm"/>
<owl:inverseOf><rdf:Property rdf:about="#narrower-term"></owl:inverseOf>
</rdf:Property>

<brdf:Property rdf:about="#narrower-term">
<rdfs:subPropertyOf rdf:resource="#hierarchical-relation"/>
</rdf:subPropertyOf>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Relation from a superordinate term to a subordinate term</rdfs:comment>
<rdfs:domain rdf:resource="#PreferredTerm"/>
<rdfs:range rdf:resource="#PreferredTerm"/>
<owl:Class rdf:parseType="Collection">
<owl:complementOf rdf:resource="#TopTerm"/>
</owl:Class>
<owl:intersectionOf rdf:resource="#PreferredTerm"/>
</owl:intersectionOf>
11.6 Testset1

<rdf:Description xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="#EXPLOSIVE">
</rdf:Description>

<rdf:Description xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" rdf:about="#SAFETY">
<oth:used-for xmlns:oth="http://www.agfa.com/w3c/2004/05OWLThesaurus#" rdf:resource="#1102"/>
11.7 Racer Test results

11.7.1 Test 1

for term X, find all terms related by equivalence-relation statement

(individual-fillers |
  http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE|
reply for test set 1
(http://www.agfa.com/w3c/2004/05OWLThesaurus#602 | http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLO)

result OK

11.7.2 Test 2
for term X, find all terms related by narrower-term
statement
reply for test set 1
(http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY)
result only the next narrower term is given, transitive property of hierarchical relation seems not to be inherited.

11.7.3 Test 3
for term X, find all terms related by broader-term
statement
reply for test set 1
(http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY)
result only the next broader term is given, transitive property of hierarchical relation seems not to be inherited.

11.7.4 Test 4
for term X, find all scope notes
statement
(individual-attribute-fillers | http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE | http://www.agfa.com/w3c/2004/05OWLThesaurus#scope-note) ; (told-value 04)
reply for test set 1
(O4), AGX-PRECIPITATIE
result OK

11.7.5 Test 5
for term X, find all assertions
statement (describe-individual | http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE)
reply for test set 1

http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE

:ASSERTIONS (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE
OTHER4)) :ROLE-FILLERS ((|
http://www.agfa.com/w3c/2004/05OWLThesaurus#ud) (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#explosiv)) (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-
relation) (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
INVESTIGATION REQUEST) (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE
http://www.agfa.com/w3c/2004/05OWLThesaurus#INAHALATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSION TEST
http://www.agfa.com/w3c/2004/05OWLThesaurus#INFLAMMATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
ANALYSIS
http://www.agfa.com/w3c/2004/05OWLThesaurus#DISPOSAL
http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMAL
STABILITY
http://www.agfa.com/w3c/2004/05OWLThesaurus#EPA
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARDOUS
http://www.agfa.com/w3c/2004/05OWLThesaurus#OSHA
http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOIGNITION
TEMPERATURE
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYREPORT
http://www.agfa.com/w3c/2004/05OWLThesaurus#EMERGENCY
http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATION
OBLIGATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARD
http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOFLAMMABIL-
ITY DETERMINATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO
http://www.agfa.com/w3c/2004/05OWLThesaurus#EINECS
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYREGULATI-
ONS
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH
POINT
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM
ADIABATIC TEMPERATURE RISE
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE-
A
http://www.agfa.com/w3c/2004/05OWLThesaurus#ALAR
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAMMABILITY
DETERMINATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYPROTECTI-
ON
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAK
http://www.agfa.com/w3c/2004/05OWLThesaurus#ACCIDENT
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYDATASHEE-
T
http://www.agfa.com/w3c/2004/05OWLThesaurus#FIRST-AID
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAC-VALUE

11/01/2012

Thesaurus representation with OWL
11.7.6 Test 6
for a preferred term X, indicate term type

statement (individual-types |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE|)

reply for testset1 (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#PreferredTerm|)

result OK

11.7.7 Test 7
for a non preferred term X, indicate term type

statement (individual-types |
http://www.agfa.com/w3c/2004/05OWLThesaurus#602|)

reply for testset1 (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#NonPreferredTerm|)

result OK

11.7.8 Test 8
find all preferred terms

statement (concept-instances |
http://www.agfa.com/w3c/2004/05OWLThesaurus#PreferredTerm|)

reply for testset1 (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM ADIABATIC TEMPERATURE RISE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARD| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#FIRST-AID| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM ADIABATIC TEMPERATURE RISE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARD| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#FIRST-AID| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARDOUS| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ALARM| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#OLP| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#INFLAMMATION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EINECS| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYREPORT| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#DISPOSAL| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAK| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH POINT| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATION OBLIGATION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EPA| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYDATASHEET| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE AFFINITY DETERMINATION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAC-VALUE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#OSHA| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSION TEST| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAMMABILITY DETERMINATION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY PROTECTION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOIGNITION TEMPERATURE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY ANALYSIS| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#INHALATION| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY REGULATIONS| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EMERGENCY| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ECOINC| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ACCIDENT| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMAL STABILITY|)

result OK

11.7.9 Test 9
find all non preferred terms

statement (concept-instances | http://www.agfa.com/w3c/2004/05OWLThesaurus#NonPreferredTerm|)

reply for test set 1 (|
http://www.agfa.com/w3c/2004/05OWLThesaurus#602| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLO| |
11.7.10 Test 10
for term X, find all terms related by hierarchical-relation

result OK

http://www.agfa.com/w3c/2004/05OWLThesaurus#1102 | http://www.agfa.com/w3c/2004/05OWLThesaurus#VEILIGHEID)

reply for test set 1
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|

http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|

http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|

http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|

http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|

http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO | |
http://www.agfa.com/w3c/2004/05OWLThesaurus#hierarchical-relation|
http://www.agfa.com/w3c/2004/05OWLThesaurus#ACCIDENT| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYDATASHEET| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#FIRST-AID| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAC-VALUE| |
http://www.agfa.com/w3c/2004/05OWLThesaurus#ECOIN| |
result  OK

11.7.11  Test 11
Find all top terms
statement  
(concept-instances |
http://www.agfa.com/w3c/2004/05OWLThesaurus#TopTerm|)
reply for test set 1  Nil
result  Not OK

11.8 Euler Test results

11.8.1 Test 1
for term X, find all terms related by equivalence-relation
statement
  # PxButton | test | Euler --think --nope --step 2000000
  DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test1 >
  test1_result_euler.n3

reply for test set 1

result  OK
11.8.2 Test 2

for term X, find all terms related by narrower-term

statement

# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2/euler/owl-rules/euler/rdfs-rules --filter test2 >
test2_result_euler.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{:ISO :narrower-term ?T} => {(?T) a :Result}.

reply for test set 1

@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix ns0: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/2000/01/rdf-schema#>.

(:SAFETY) a :Result.

result OK

11.8.3 Test 3

for term X, find all terms related by broader-term

statement

# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2/euler/owl-rules/euler/rdfs-rules --filter test3 >
test3_result_euler.n3

# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3/
euler/rdfs-rules.n3 --think --filter=test3.n3 >
test3_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{:EXPLOSIVE :broader-term ?T} => {(?T) a :Result}.

reply for testset1

@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix nso: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

(:SAFETY) a :Result.

result only the next broader term is given, transitive property of hierarchical relation seems not to be inherited.

11.8.4 Test 4
for term X, find all scope notes

statement
  # PxButton | test | Euler --think --nope --step 2000000
  DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test4 >
  test4_result_euler.n3

reply for testset1
@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix nso: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{"EXPLOSIVE :scope-note ?T} => {(?T) a :Result}.

("AGX-PRECIPITATIE") a :Result.

result OK

11.8.5 Test 5
for term X, find all assertions

statement
(describe-individual # PxButton | test | Euler --think --nope --step 2000000 DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test5 > test5_result_euler.n3
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.


reply for testset1
@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix ns0: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

(:associative-relation :OLP) a :Result.
(:broader-term :SAFETY) a :Result.
(:equivalence-relation :602) a :Result.
(:equivalence-relation :EXPLO) a :Result.
(:hierarchical-relation :SAFETY) a :Result.
(:hierarchical-relation :EXPLOSIVE A) a :Result.
(:hierarchical-relation :ACCIDENT) a :Result.
(:hierarchical-relation :ALARM) a :Result.
(:hierarchical-relation :DISPOSAL) a :Result.
(:hierarchical-relation :DUST TEST) a :Result.
(:hierarchical-relation :ECOIN) a :Result.
(:hierarchical-relation :EINECS) a :Result.
(:hierarchical-relation :EMERGENCY) a :Result.
(:hierarchical-relation :HAZARD) a :Result.
(:hierarchical-relation :HAZARDOUS) a :Result.
(:hierarchical-relation :INFLAMMATION) a :Result.
(:hierarchical-relation :INHALATION) a :Result.
(:hierarchical-relation :MAK) a :Result.

result OK

11.8.6 Test 6
for a preferred term X, indicate term type
statement

(individual-types # PxButton | test | Euler --think --nope --step 2000000 DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test6 > test6_result_euler.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{?C a owl:Class. :EXPLOSIVE a ?C} => {(?C) a :Result}.

reply for testset1

@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix ns0: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>. 

11/01/2012
Thesaurus representation with OWL
113 / 143
11.8.7 Test 7
for a non preferred term X, indicate term type

\[
\text{statement} = \text{PxButton} \mid \text{test} \mid \text{Euler} --\text{think} --\text{nope} --\text{step} 2000000 \\
\text{DE_testset_1_2} /\text{euler/owl-rules} /\text{euler/rdfs-rules} --\text{filter test7} > \\
\text{test7_result_euler.n3}
\]

\[
@\text{prefix owl:} <\text{http://www.w3.org/2002/07/owl#}>. \\
@\text{prefix rdfs:} <\text{http://www.w3.org/2000/01/rdf-schema#}>. \\
@\text{prefix rdf:} <\text{http://www.w3.org/1999/02/22-rdf-syntax-ns#}>. \\
@\text{prefix :} <\text{http://www.agfa.com/w3c/2004/05OWLThesaurus#}>
\]

\[
\{?C \text{ a owl:Class.} :602 \text{ a } ?C} \Rightarrow \{(?C) \text{ a :Result}\}
\]

\[
\text{reply for testset1} = @\text{prefix log:} <\text{http://www.w3.org/2000/10/swap/log#}>. \\
@\text{prefix ns1:} <\text{http://www.agfa.com/w3c/euler/rdfs-rules#}>. \\
@\text{prefix str:} <\text{http://www.w3.org/2000/10/swap/string#}>. \\
@\text{prefix owl:} <\text{http://www.w3.org/2002/07/owl#}>. \\
@\text{prefix ns0:} <\text{http://www.agfa.com/w3c/euler/owl-rules#}>. \\
@\text{prefix rdf:} <\text{http://www.w3.org/1999/02/22-rdf-syntax-ns#}>. \\
@\text{prefix XML:} <\text{http://www.w3.org/2001/XMLSchema#}>. \\
@\text{prefix :} <\text{http://www.agfa.com/w3c/2004/05OWLThesaurus#}>
\]

\[
\text{:NonPreferredTerm} \text{ a :Result.}
\]

11.8.8 Test 8
find all preferred terms

\[
\text{statement} = \text{PxButton} \mid \text{test} \mid \text{Euler} --\text{think} --\text{nope} --\text{step} 2000000 \\
\text{DE_testset_1_2} /\text{euler/owl-rules} /\text{euler/rdfs-rules} --\text{filter test8} > \\
\text{test8_result_euler.n3}
\]

\[
@\text{prefix owl:} <\text{http://www.w3.org/2002/07/owl#}>. \\
@\text{prefix rdfs:} <\text{http://www.w3.org/2000/01/rdf-schema#}>. \\
@\text{prefix rdf:} <\text{http://www.w3.org/1999/02/22-rdf-syntax-ns#}>. \\
@\text{prefix :} <\text{http://www.agfa.com/w3c/2004/05OWLThesaurus#}>
\]

\[
\{?T \text{ a :PreferredTerm} \} \Rightarrow \{(?T) \text{ a :Result}\}.
\]

\[
\text{reply for testset1} = @\text{prefix log:} <\text{http://www.w3.org/2000/10/swap/log#}>. \\
@\text{prefix ns1:} <\text{http://www.agfa.com/w3c/euler/rdfs-rules#}>. \\
@\text{prefix str:} <\text{http://www.w3.org/2000/10/swap/string#}>
\]
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix nso: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

(:ACCIDENT) a :Result.
(:ALARM) a :Result.
(:DISPOSAL) a :Result.
(:ECOIN) a :Result.
(:EINECS) a :Result.
(:EMERGENCY) a :Result.
(:EPA) a :Result.
(:FIRST-AID) a :Result.
(:HAZARD) a :Result.
(:HAZARDOUS) a :Result.
(:INFLAMMATION) a :Result.
(:INHALATION) a :Result.
(:MAC-VALUE) a :Result.
(:MAK) a :Result.
(:OSHA) a :Result.
((http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYANALYSIS>) a :Result.
(:SAFETYDATASHEET) a :Result.
(:SAFETYPROTECTION) a :Result.
(:SAFETYREGULATIONS) a :Result.
(:SAFETYREPORT) a :Result.
(<http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMAL STABILITY>) a :Result.
(:\OLP) a :Result.
(:\ISO) a :Result.
(:\EXPLOSIVE) a :Result.
(:\SAFETY) a :Result.
(<http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE A>) a :Result.

result          OK ?

11.8.9 Test 9
find all non preferred terms
statement
# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test9 >
test9_result_euler.n3
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{"T a :NonPreferredTerm} => {(?T) a :Result},
reply for testset1

@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix ns0: <http://www.agfa.com/w3c/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.

(:\602) a :Result.
(:\EXPLO) a :Result.
(:\1102) a :Result.
(:\VEILIGHEID) a :Result.
result          OK

11.8.10 Test 10
for term X, find all terms related by hierarchical-relation
statement
# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test10 >
test10_result_euler.n3
reply for testset1

{:SAFETY :hierarchical-relation ?T} => {(?T) a :Result}.

(:ISO) a :Result.

(:SAFETY) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIV E A>) a :Result.

(:ACCIDENT) a :Result.

(:ALARM) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOFLA MIBILITY DETERMINATION>) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOIGNI TION TEMPERATURE>) a :Result.

(:DISPOSAL) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST>) a :Result.

(:ECOIN) a :Result.

(:EINECS) a :Result.

(:EMERGENCY) a :Result.

(:EPA) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIO N TEST>) a :Result.

(:EXPLOSIVE) a :Result.

(:FIRST-AID) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAMMAB ILITY DETERMINATION>) a :Result.

(<http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH POINT>) a :Result.

(:HAZARD) a :Result.

(:HAZARDOUS) a :Result.

(:INFLAMMATION) a :Result.

(:INHALATION) a :Result.

(:MAC-VALUE) a :Result.

(:MAK) a :Result.
(«http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM
ADIABATIC TEMPERATURE RISE») a :Result.
(«http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATION
OBLIGATION») a :Result.
(:OSHA) a :Result.
(«http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
ANALYSIS») a :Result.

result OK?

11.8.11 Test 11
Find all top terms

statement

# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test11 >
test11_result_euler.n3
# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test11.n3 >
test11_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{?T a :TopTerm} => {?T} a :Result).

reply for testset1

@prefix log: <http://www.w3.org/2000/10/swap/log#>.
@prefix ns1: <http://www.agfa.com/w3c/euler/rdfs-rules#>.
@prefix str: <http://www.w3.org/2000/10/swap/string#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix ns0: <http://www.agfa.com/w3c/euler/owl-rules#>.
@prefix rdf: <http://www.w3.org/1999/02/rdf-syntax-ns#>.
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix rdf: <http://www.w3.org/2001/XMLSchema#>

result Not OK

11.9 CWM Testresults

11.9.1 Test 1
for term X, find all terms related by equivalence-relation

statement

# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test11.n3 > test11_result_cwm.n3
reply for testset1

# Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{:EXPLOSIVE :equivalence-relation ?T} => {(?T) a :Result}.

result  OK

11.9.2 Test 2

for term X, find all terms related by narrower-term statement

# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test2.n3 >
test2_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{:ISO :narrower-term ?T} => {(?T) a :Result}.

reply for testset1

# Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
result not OK

The narrower terms from ISO are not explicitly stated in testset 1. A non OWL reasoner can not infer this because the inverse-of property of narrower-term is a OWL specific feature. This will be solved when the complete thesaurus is translated to RDFS/XML, then this statement will be explicitly available.

11.9.3 Test 3

for term X, find all terms related by broader-term

statement

```python
# PxBUTTON | CWM | PYTHON
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test3.n3 > test3_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{:EXPLOSIVE :broader-term ?T} => {(?T) a :Result}.
```

reply for testset1

```
# Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
@prefix :
<http://www.agfa.com/w3c/2004/05OWLThesaurus#> .

( :SAFETY )
a :Result .
```

result OK

11.9.4 Test 4

for term X, find all scope notes

statement

```
# PxBUTTON | CWM | PYTHON
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test4.n3 > test4_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
```
reply for testset1
# Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#> .

( "AGX-PRECIPITATIE"^^XML:string )
a :Result .

#ENDS

result

OK

11.9.5 Test 5
for term X, find all assertions

statement
# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test5.n3 >
test5_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.


reply for testset1
# Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#> .
@prefix XML: <http://www.w3.org/2001/XMLSchema#> .

(:associative-relation
 :OLP )
a :Result .
(:broader-term
 :SAFETY )
a :Result .
(:equivalence-relation
 :602 )
a :Result .
(:equivalence-relation
 :EXPLO )
a :Result .
(:hierarchical-relation
 <http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLO SIVE A> )
a :Result .
(:hierarchical-relation
 :SAFETY )
a :Result .
(:narrower-term
 <http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLO SIVE A> )
a :Result .
(:related-term
 :OLP )
a :Result .
(:scope-note
 "AGX-PRECIPITATIE"^^XML:string )
a :Result .
(:used-for
 :602 )
a :Result .
(:used-for
 :EXPLO )
a :Result .

#ENDS

result OK?

11.9.6 Test 6
for a preferred term X, indicate term type

statement

# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test6.n3 >
test6_result_cwm.n3
for a non preferred term X, indicate term type

```
Test 7
```

```python
for a non preferred term X, indicate term type
```
11.9.8 Test 8

find all preferred terms

statement

# PxButton | test | Euler --think --nope --step 2000000
DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test8 >
test8_result_euler.n3
# PxButton | cwm | python
/w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test8.n3 >
test8_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{?T a :PreferredTerm} => {(?T) a :Result}.

reply for testset1

#Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl
Exp

# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
@prefix :
<http://www.agfa.com/w3c/2004/05OWLThesaurus#> .

( :ACCIDENT )
a :Result .
( :ALARM )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOF LAMMABILITY DETERMINATION>  )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOI GNITION TEMPERATURE>  )
a :Result .
( :DISPOSAL )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST>  )
a :Result .
( :ECOIN )
a :Result .
( :EINECS )
a :Result .
( :EMERGENCY )
a :Result .
( :EPA )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSION TEST>  )
a :Result .
( :EXPLOSIVE )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE A>  )
a :Result .
( :FIRST-AID )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAM MABILITY DETERMINATION>  )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH POINT>  )
a :Result .
( :HAZARD )
a :Result .
( :HAZARDOUS )
a :Result .
( :INFLAMMATION )
a :Result .
( :INHALATION )
a :Result .
( :ISO )
a :Result .
( :MAC-VALUE )
a :Result .
( :MAK )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUMADIABATICTEMPERATURERISE> )
   a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATIONOBLIGATION> )
   a :Result .
( :OLP )
   a :Result .
( :OSHA )
   a :Result .
( :SAFETY )
   a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYANALYSIS> )
   a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYINVESTIGATIONREQUEST> )
   a :Result .
( :SAFETYDATASHEET )
   a :Result .
( :SAFETYPROTECTION )
   a :Result .
( :SAFETYREGULATIONS )
   a :Result .
( :SAFETYREPORT )
   a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMALSTABILITY> )
   a :Result .

#ENDS

result          OK

11.9.9 Test 9
find all non preferred terms
statement
   # PxButton | test | Euler --think --nope --step 2000000
   DE_testset_1_2 /euler/owl-rules /euler/rdfs-rules --filter test9 >
   test9_result_euler.n3
   # PxButton | cwm | python
   /w3ccvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
   /euler/rdfs-rules.n3 --think --filter=test9.n3 >
   test9_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.
{?T a :NonPreferredTerm} => {(?T) a :Result}.

reply for testset1

#Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
# using base file:/2004/05OWLThesaurus/

# Notation3 generation by
# notation3.py,v 1.155 2004/06/25 01:26:59 timbl Exp

# Base was: file:/2004/05OWLThesaurus/
@prefix :
<http://www.agfa.com/w3c/2004/05OWLThesaurus#> .

( :1102 )
a :Result .
( :602 )
a :Result .
( :EXPLO )
a :Result .
( :VEILIGHEID )
a :Result .

#ENDS
result OK

11/01/2012 Thesaurus representation with OWL 127 / 143
@prefix :<http://www.agfa.com/w3c/2004/05OWLThesaurus#> .

( :ACCIDENT )
a :Result .
( :ALARM )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOLUMMABILITY DETERMINATION> )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOMATIC IGNITION TEMPERATURE> )
a :Result .
( :DISPOSAL )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST> )
a :Result .
( :ECOIN )
a :Result .
( :EINECS )
a :Result .
( :EMERGENCY )
a :Result .
( :EPA )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSION TEST> )
a :Result .
( :EXPLOSIVE )
a :Result .
( :FIRST-AID )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAMMABILITY DETERMINATION> )
a :Result .
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH POINT> )
a :Result .
( :HAZARD )
a :Result .
( :HAZARDOUS )
a :Result .
( :INFLAMMATION )
a :Result .
( :INHALATION )
a :Result .
( :ISO )
a :Result .
( :MAC-VALUE )
a :Result.
( :MAK )
a :Result.
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM ADIABATIC TEMPERATURE RISE> )
a :Result.
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATION OBLIGATION> )
a :Result.
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY ANALYSIS> )
a :Result.
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST> )
a :Result.
( :SAFETYDATASHEET )
a :Result.
( :SAFETYPROTECTION )
a :Result.
( :SAFETYREGULATIONS )
a :Result.
( :SAFETYREPORT )
a :Result.
( <http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMAL STABILITY> )
a :Result.

#ENDS

result OK

11.9.11 Test 11
Find all top terms

statement

# PxButton | cwm | python
/w3cvs/WWW/2000/10/swap/cwm.py DE_testset_1_2.n3
/euler/rdfs-rules.n3 --think --filter=test11.n3 >
test11_result_cwm.n3

@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://www.agfa.com/w3c/2004/05OWLThesaurus#>.

{?T a :TopTerm} => {?T a :Result}.

reply for testset1

#Processed by Id: cwm.py,v 1.154 2004/06/24 03:17:11 timbl Exp
result Not OK but expected for a RDF query engine

11.10 Sesame Test results

11.10.1 Test 1
for term X, find all terms related by equivalence-relation

statement
SELECT C
FROM {<oth:EXPLOSIVE>} <oth:equivalence-relation> {C}
using namespace
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  rdfs = <http://www.w3.org/2000/01/rdf-schema#>,
  owl = <http://www.w3.org/2002/07/owl#>,
  oth = <http://www.agfa.com/w3c/2004/05OWLThesaurus#>

reply for testset1
Query results: C
http://www.agfa.com/w3c/2004/05OWLThesaurus#602
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLO
2 results found in 90 ms.

result OK

11.10.2 Test 2
for term X, find all terms related by narrower-term

statement
SELECT C
FROM {<oth:ISO>} <oth:narrower-term> {C}
using namespace
  rdf = <http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
  rdfs = <http://www.w3.org/2000/01/rdf-schema#>,
  owl = <http://www.w3.org/2002/07/owl#>,
  oth = <http://www.agfa.com/w3c/2004/05OWLThesaurus#>

reply for testset1 None
result NOK

The narrower terms from ISO are not explicitly stated in testset 1.
A non OWL reasoner cannot infer this because the inverse-of property of
narrower-term is an OWL specific feature. This will be solved when the
complete thesaurus is translated to RDFS/XML, then this statement will
be explicitly available.
11.10.3 Test 3
for term X, find all terms related by broader-term statement

```
SELECT C
FROM {<oth:EXPLOSIVE>} <oth:broader-term> {C}
```

using namespace
```
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>
```

reply for testset1
Query results:
```
C
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
```
1 results found in 40 ms.

result only the next broader term is given, transitive property of hierarchical relation is a OWL feature.

11.10.4 Test 4
for term X, find all scope notes statement

```
SELECT C
FROM {<oth:EXPLOSIVE>} <oth:scope-note> {C}
```

using namespace
```
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>
```

reply for testset1 Query results:
```
C
"AGX-PRECIPITATIE"^^http://www.w3.org/2001/XMLSchema#string
```
1 results found in 20 ms.

result OK

11.10.5 Test 5
for term X, find all assertions statement

```
SELECT D,C
FROM {<oth:EXPLOSIVE>} D {C}
```

using namespace
```
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>
```

reply for testset1
Query results:
Sesame is not aware of the OWL specific features, transitivity of certain properties is not applied resulting in a smaller result set, 1 level deep compared to the OWL reasoners. This could be solved to iteratively make new queries with the output as input.

11.10.6 Test 6

for a preferred term X, indicate term type

**statement**

```
SELECT C
FROM {<oth:EXPLOSIVE>} <serql:directType> {C}
```

using namespace
```
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
```

"AGX-PRECIPITATIE"^^http://www.w3.org/2001/XMLSchema#string
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>

**reply for testset1** Query results:

```
C
http://www.agfa.com/w3c/2004/05OWLThesaurus#PreferredTerm
```

2 results found in 10 ms.

**result**  OK

11.10.7  Test 7

for a non preferred term X, indicate term type

**statement**

```
SELECT C
FROM {<oth:602>} <serql:directType> {C}
using namespace
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
other = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>
```

**reply for testset1** Query results:

```
C
```

```
http://www.agfa.com/w3c/2004/05OWLThesaurus#NonPreferredTerm
```

1 result found in 10 ms.

**result**  OK

11.10.8  Test 8

find all preferred terms

**statement**

```
SELECT C
FROM {C} <rdf:type> {<oth:PreferredTerm>}
using namespace
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
other = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>
```

**reply for testset1** Query results:

```
C
```

```
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
http://www.agfa.com/w3c/2004/05OWLThesaurus#OLP
http://www.agfa.com/w3c/2004/05OWLThesaurus#ISO
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSIVE A
http://www.agfa.com/w3c/2004/05OWLThesaurus#ACCIDENT
http://www.agfa.com/w3c/2004/05OWLThesaurus#ALARM
http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOFLAMMABILITY DETERMINATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#AUTOIGNITION TEMPERATURE
```

11/01/2012
http://www.agfa.com/w3c/2004/05OWLThesaurus#DISPOSAL
http://www.agfa.com/w3c/2004/05OWLThesaurus#DUST TEST
http://www.agfa.com/w3c/2004/05OWLThesaurus#ECOIN
http://www.agfa.com/w3c/2004/05OWLThesaurus#EINECS
http://www.agfa.com/w3c/2004/05OWLThesaurus#EMERGENCY
http://www.agfa.com/w3c/2004/05OWLThesaurus#EPA
http://www.agfa.com/w3c/2004/05OWLThesaurus#EXPLOSION TEST
http://www.agfa.com/w3c/2004/05OWLThesaurus#FIRST-AID
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLAMMABILITY DETERMINATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#FLASH POINT
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARD
http://www.agfa.com/w3c/2004/05OWLThesaurus#HAZARDOUS
http://www.agfa.com/w3c/2004/05OWLThesaurus#INFLAMMATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#INHALATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAC-VALUE
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAK
http://www.agfa.com/w3c/2004/05OWLThesaurus#MAXIMUM ADIABATIC TEMPERATURE RISE
http://www.agfa.com/w3c/2004/05OWLThesaurus#NOTIFICATION OBLIGATION
http://www.agfa.com/w3c/2004/05OWLThesaurus#OSHA
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY ANALYSIS
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY INVESTIGATION REQUEST
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYDATASHEET
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYPROTECTION
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYREGULATIONS
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETYREPORT
http://www.agfa.com/w3c/2004/05OWLThesaurus#THERMAL STABILITY
35 results found in 10 ms.

result OK

11.10.9 Test 9
find all non preferred terms

statement
SELECT C
FROM {C} <rdf:type> {<oth:NonPreferredTerm>}
using namespace
rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>

reply for testset1 Query results:
C
for term X, find all terms related by hierarchical-relation

statement

SELECT D
FROM {<oth:ISO>} <oth:hierarchical-relation> {D}
using namespace
   rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
   rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
   owl = <!http://www.w3.org/2002/07/owl#>,
   oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>

reply for testset1

0 results

result NOK

This result is expected because there is not statement about with ISO as subject and a hierarchical relation as property present in testset1. There are statements were ISO is the object and hierarchical-relation the property but inverse-of is a OWL feature and is not used by Sesame.

The next query:
SELECT D
FROM {D}  <oth:hierarchical-relation> {<oth:ISO>}
using namespace
   rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
   rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
   owl = <!http://www.w3.org/2002/07/owl#>,
   oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>

gives as result:
Query results:
D
http://www.agfa.com/w3c/2004/05OWLThesaurus#SAFETY
1 results found in 10 ms.
Which is correct but not complete because the transitive property of the hierarchical-relation can not be interpreted by Sesame.

Find all top terms

statement

SELECT D
FROM {D} <rdf:type> {<oth:TopTerm>}
using namespace
   rdf = <!http://www.w3.org/1999/02/22-rdf-syntax-ns#>,
   rdfs
rdfs = <!http://www.w3.org/2000/01/rdf-schema#>,
owl = <!http://www.w3.org/2002/07/owl#>,
oth = <!http://www.agfa.com/w3c/2004/05OWLThesaurus#>

**reply for testset1** Nil

**result** Not OK but expected, only OWL reasoner could infer this information
12 Reference List

4. Conceptual relationships for encoding thesauri, classification systems and organized metadata collections and a proposal for encoding a core set of thesaurus relationships using RDF Schema. 2000. Ref Type: Generic
8. RDF Thesaurus Wiki. SWAD - Europe. 17-11-2003c. Ref Type: Internet Communication
9. The CERES/NBII thesaurus. 2-12-2003d. Ref Type: Generic
10. Gateway to Educational Materials. 2004a. Ref Type: Generic
12. Taxonomy. Wikipedia, the free encyclopedia. 2004c. Ref Type: Generic
14. A.Ouksel and A.Sheth. Special section on semantic interoperability in global information systems: a brief introduction to the research area on the special section. SIGMOD Record 28[1]. 1999. Ref Type: Generic
15. Alistair J.Miles, Nikki Rogers, and Dave Beckett. SKOS-Core 1.0 Guide. An RDF Schema for thesauri and related knowledge organisation systems. 4-5-2004. Ref Type: Generic
Ref Type: Generic

Ref Type: Thesis/Dissertation

Ref Type: Report

Ref Type: Generic


Ref Type: Generic


Ref Type: Generic

Ref Type: Generic

Ref Type: Generic

Ref Type: Generic

Ref Type: Generic


52. P.M. Roget. ROGET’S THESAURUS OF ENGLISH WORDS AND PHRASES. 1911. Ref Type: Data File
54. Phil Cross, Dan Brickley, and Traugott Koch. RDF thesaurus specification. 2000. The Institute for Learning and Research Technology, University of Bristol. Ref Type: Report
55. R. Cornet. RICE, the Racer Interactive Client Environment. 2004. Ref Type: Generic
62. Tim Berners-Lee. CWM - Closed World Machine. 5-7-2001a. Ref Type: Generic
Ref Type: Generic
Ref Type: Slide
CDS Communications Data Services.
Ref Type: Generic
Ref Type: Generic
Ref Type: Generic
Ref Type: Generic