D2.5.4 Analysis of Requirements for Further Language Extensions

Giorgos Stoilos, Giorgos Stamou (ITI/CERTH)
Rob Searer, Ian Horrocks (UoM), Jeff Z. Pan (UoA), Alex Polleres (UIBK), Holger Lausen (NUIG), Mustafa Jarrar (VUB), Carlos F. Enguix (NUIG), Holger Wache (VU), York Sure (UKARL)

Abstract.
EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB Deliverable D2.5.4 (WP2.5)
Keyword list: description logics, ontology language, query language, RDF, OWL DL, OWL-E
The current deliverable surveys a number of real-world use cases encountered in the course of the KnowledgeWeb project which demonstrate limitations of the Semantic Web ontology languages RDF and OWL. Approaches to overcoming these limitations are considered, including increased expressiveness of semantic languages, additional or alternative semantic models, and usability features which make ontologies easier to work with in real-world software architectures. Potential impacts of these approaches on the use cases presented here, as well as on other use cases including those identified by industrial work packages, are described. The issues raised by the use cases considered will play an important role in the future for the use of ontology languages in many applications. This deliverable builds upon the work presented in deliverables 2.5.1, 2.5.2, and 2.5.3, which presented RDF and OWL extensions for the representation of rules, queries, extended datatypes, uncertainty, and context, as well as studies on implementation and optimization issues.

Document Identifier	KWEB/2004/D2.5.4/v0.4
Project	KWEB EU-IST-2004-507482
Version	0.11
Date	
State	draft
Distribution	public

Knowledge Web Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) - Coordinator
Institute of Computer Science
Technikerstrasse 13
A-6020 Innsbruck
Austria
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polytechnique Fédérale de Lausanne (EPFL)
Computer Science Department
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne
Switzerland
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse 9
14195 Berlin
Germany
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe -
Montbonnot Saint Martin
38334 Saint-Ismier
France
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi - Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland
Science and Technology Building
University Road
Galway
Ireland
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA
United Kingdom
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale Beschreibungsverfahren - AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany
Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of Computer Science, University of Manchester, Oxford Road
Manchester, M13 9PL
United Kingdom
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to writing parts of this document:

Centre for Research and Technology Hellas /Informatics and Telematics Institute
Free University of Bozen-Bolzano
Institut National de Recherche en Informatique et en Automatique
University of Manchester
University of Trento
Vrije Universiteit Amsterdam
University of Innsbruck

Changes

Version	Date	Author(s)	Changes
0.1	09.09.05	Giorgos Stoilos, Giorgos Stamou	creation
0.2	06.10.05	Giorgos Stoilos, Giorgos Stamou	Uncertainty Handling Requirement (1st version)
0.3	20.11.05	Axel Polleres, Holger Lausen	Requirements for Semantic Web Service Description and Information Integration (1st version)
0.4	24.11.05	Jeff Pan	Requirements for user-defined datatypes and datatype predicates
0.5	25.11.05	Mustafa Jarrar	Requirements for Ontology Engineering in Business Applications
0.6	25.11.05	Carlos F. Enguix	Requirements for Ontology Versioning
0.7	30.11.05	Holger Wache, York Sure	Requirements for Ontology Modularization and Alignment
0.8	30.11.05	Giorgos Stoilos, Giorgos Stamou	Uncertainty Handling Requirement (2nd version)
0.9	04.12.05	Rob Shearer	Requirements for Brain Anatomy and Protein classification
0.10	06.12.05	Rob Shearer	editing
0.11	07.12.05	Giorgos Stoilos	editing

Executive Summary

In Deliverables 2.5.1, 2.5.2 and 2.5.3, various extensions of the Semantic Web ontology languages, namely RDF and OWL, have been proposed covering issues such as extended datatypes, uncertainty, and context, as well as providing formal representations for rules and queries. Studies on implementation and optimization issues for these extensions have also been provided. In the current deliverable, a number of use cases which have been encountered in practice and demonstrate limitations of the existing Semantic Web languages are described. Approaches to overcoming these limitations are considered, including increased expressiveness of semantic languages, additional or alternative semantic models, and usability features which make ontologies easier to work with in real-world software architectures. Potential impacts of these approaches on the use cases presented here, as well as on other use cases including those identified by industrial work packages, are described. The issues raised by the use cases considered will play an important role in the future for the use of ontology languages in many applications.

Contents

Chapter 1

Introduction

According to widely known proposals for a Semantic Web architecture, ontologies will play a key role as they will be used as a source of shared and precisely defined terms that can be used in metadata [LR99]. This has led to considerable efforts to develop a suitable ontology language, culminating in the design of the OWL Web Ontology Language [BvHH$^+$04], which is now a W3C recommendation. The OWL recommendation actually consists of three languages of increasing expressive power: OWL Lite, OWL DL and OWL Full. *OWL Lite* and *OWL DL*[1] are, like DAML+OIL, basically very expressive description logics; they are almost[2] equivalent to the $\mathcal{SHIF}(\mathbf{D}^+)$ and $\mathcal{SHOIN}(\mathbf{D}^+)$ DLs. OWL Full is clearly undecidable because it does not impose restrictions on the use of transitive properties; furthermore, it has been argued that its semantics are problematic [PH03a].

Although OWL adds considerable expressive power with respect to languages such as RDF, it does have several limitations to expressiveness when used for modelling knowledge in several domains of interest. For example, OWL does not provide a composition constructor, so it is impossible to capture relationships between a composite property and another (possibly composite) property. Further, it has a very weak datatype system, which does not provide a framework for user-defined datatypes. These and several other weaknesses of the OWL ontology language have been identified by several scientists and have led to a number of proposed extensions.

Several of these extensions have been investigated or proposed in previous deliverables. Furthermore, their reasoning and implementation capabilities have also been studied and detailed algorithms have been proposed. In the current deliverable we step back to consider some representative use cases encountered in practice. Approaches to overcoming these limitations (described by specific requirements) are considered, including including increased expressiveness of semantic languages, additional or alternative semantic models, and usability features which make ontologies easier to work with in real-

[1]'DL' for Description Logic

[2]They also provide annotation properties, which Description Logics do not.

world software architectures. Such approaches are expected to benefit the provided use cases as well as generic ontology engineering capabilities, including evolution, modularization, alignment, and semantic web services.

1.1 Existing OWL extensions

Several extensions to Semantic Web ontology languages have been proposed during the last decade. In the current section we try to summarize the most important and widely studied ones.

As we mentioned in the previous section, OWL lacks a composition constructor, hence few things can be said regarding properties. The standard example here is the obvious relationship between the composition of the "parent" and "brother" properties and the "uncle" property. One way to address this problem would be to extend OWL with some form of "rules language". One such proposed extension is SWRL (Semantic Web Rule Language) [HPSB$^+$04], which is a Horn clause rules extension to OWL DL that overcomes many of these limitations. SWRL extends OWL in a syntactically and semantically coherent manner: the basic syntax for SWRL rules is an extension of the abstract syntax for OWL DL and OWL Lite; SWRL rules are given formal meaning via an extension of the OWL DL model-theoretic semantics; SWRL rules are given an XML syntax based on the OWL XML presentation syntax; and a mapping from SWRL rules to RDF graphs is given based on the OWL RDF/XML exchange syntax. However, the straightforward combination of OWL DL with SWRL is undecidable and a thorough analysis of the decidable fragments of such an OWL DL+SWRL combination (analogous to those which have been performed for the description logics component of OWL) is still missing and not reflected in the current language definition.

Another key extension of OWL is a query language that provides a formalism for agents to query information stored in (possibly multiple) OWL *knowledge bases* (or simply *KB*), consisting of (possibly multiple) sets of OWL statements. The Joint US/EU ad hoc Agent Markup Language Committee[3] has proposed an OWL query language called OWL-QL [FHH03], as a candidate standard language, which is a direct successor of the DAML Query Language (DQL) [FHe03], also released by the Joint US/EU ad hoc Agent Markup Language Committee. Both language specifications go beyond the aims of other current web query languages like XML Query [BCF$^+$03], an XML [BPSM$^+$04] query language, or RQL [KAC$^+$02] and SPARQL [Pe04], two RDF [Bec04] query languages, in that they support the use of inference and reasoning services for query answering.

Although OWL adds considerable expressive power to the Semantic Web, the OWL datatype formalism (or simply *OWL datatyping*) is much too weak for many applications. E.g., OWL datatyping does not provide a general framework for user-defined datatypes, such as XML Schema derived datatypes, nor does it support n-ary datatype predicates

[3] See `http://www.daml.org/committee/` for the members of the Joint Committee.

(such as the binary predicate $>$ for integers), not to mention user-defined datatype predicates (such as the binary predicate $>$ for non-negative integers). For that purpose, the datatype group approach is presented in [PH03b], in order to extend OWL datatyping with datatype predicates.

Uncertainty reasoning extensions have gained considerable attention over the last decade. This is mainly due to the fact that several domains and applications deal with information that is inherently imprecise or vague. For example, decision making, image and video processing, medical diagnosis, robotics, and information retrieval are all examples of applications for which special mathematical frameworks have been adopted to cope with uncertain information. Such approaches can be divided into probabilistic extensions [GL02, DP04], possibilistic extensions [Ber94] and fuzzy extensions [Str01, SST+05a, SST+05b]. Different extensions aim to cover different types of uncertainty. More precisely, probability covers randomness, possibility theory aims at covering imprecision, while fuzzy extensions aim at covering vagueness as well as formalizing possibility theory [KY95].

Finally, in order to achieve semantic interoperability and inter-connectivity between different OWL ontologies, *context* extensions to the OWL language such as C-OWL have been proposed [BGvH+03]. C-OWL allows spaces of heterogeneous ontologies to be related via a set of formal semantic mappings, also called contextualized ontologies.

Most of the above extensions and their implementation capabilities have been investigated in previous deliverables [PFT+04a, PFT+05].

1.2 Reader's Guide

The rest of the deliverable is organized as follows. Chapter 2 presents several real usage scenarios which have been encountered in the ontology engineering domain while exploring ontology alignment, evolution, modularization and more. Each usage scenario demonstrates one or more limitation of the current OWL language. In Chapter 3, several specific approaches are provided which could overcome the identified limitations and support the use cases presented in the previous chapter as well as other potential use cases, in particular those identified by industrial members as a part of deliverable 1.1.4 [LN05]. Finally, Chapter 4 summarizes our findings.

Chapter 2

Use Cases

In this chapter, a number of real-world usage scenarios are described. Each usage scenario demonstrates some limitation of the OWL and RDF langauges. In some cases such limitations (and potential remedies) are evident from brief overviews of the application, while other scenarios must be described in more detail in order to identify the source of implementation problems. This is particularly true of applications whose specific goals predate the decision to make use of OWL and RDF during implementation—since no fixed semantic architecture is dictated, both the languages themselves and the usage patterns for those languages within the overall system must be considered.

Specific language limitations (in terms of both expressiveness and usability) as perceived by implementors are identified for each usage scenario. It is worth noting that the identification of issues does not suggest that OWL and RDF are unsuitable for the applications described, or that there are not other usage patterns which could exploit these languages in order to achieve the applications' goals. Rather, such scenarios should be viewed as opportunities found by implementors to extend and improve semantic applications, preferably using tose applications' existing architectures, which could exist if certain limitions of semantic web languages could be overcome. In Chapter 3, approaches to overcoming some of these limitations will be presented.

2.1 Brain Anatomy

2.1.1 Overview

The general framework is sharing anatomical knowledge (ontology and rules) and tools (services) needed in the context of neuroimaging, applied both to medical practice, i.e. decision support in neurology and neurosurgery, and to research about neurological pathology such as epilepsy, dementia, etc. The application aims at developing new methods for assisting the labeling of the brain cortex structures - sulci and gyri - in MRI images.

Indeed, the brain cortex can nowadays be automatically segmented as a whole but the problem remains to identify its various parts. Numerical tools previously developed at IDM provide a list of items corresponding to the gyrus parts and sulcus segments separating them, recognized in the images. Each item is associated with a set of features: (1) attributes depicting intrinsic properties, such as the length and depth of a sulcus segment, or the surface of a gyrus part, (2) binary relationships, such as the neighborhood of two gyrus parts, the connection of two sulcus segments, (3) n-ary relationships such as the separation of two gyrus parts by a sulcus segment. However, as they are generated by numerical tools, such items are unlabeled. The approach proposed to assist their labeling relies on a brain ontology storing the a priori ÒcanonicalÓ knowledge about the most important sulci and gyri, and on a rule base describing the dependencies between the properties of the brain cortex structures. Documentation about the ontology and the rules was prepared for the W3C Workshop on Rule Languages for Interoperability [GBDG05] and is available at

`http://idm.univ-rennes1.fr/~obierlai/anatomy/annexes/index.html`

.

2.1.2 Limitations of existing languages

We need for this application an Ontology language that offers OWL DL expressiveness, extended by qualified cardinality constraints. OWL DL expressiveness is at least needed ($\exists$ and $\sqcup$ occur in class definitions). OWL DL was selected to benefit of DL reasoning services (consistency checking and automatic classification) and because OWL DL reasoners are available e.g. Racer, Pellet3. Its extension by qualified cardinality constraints would be particularly useful in anatomy for defining structures from their parts, or from their boundaries, or combinations of both. For example, they are needed to represent in OWL-DL a 'hemisphere' as an anatomical entity whose direct parts are lobes, each part being of a distinct type (i.e. frontal lobe, parietal lobe, occipital lobe, limbic lobe, temporal lobe), or similarly to express that a precentral gyrus is bounded by exactly one precentral sulcus, one central sulcus, and is connected or contiguous to one postcentral gyrus. Additionally, we need OWL DL to be extended by a Web rule language that offers at least Datalog rules. DL extensions such as SHIQ added with Role Inference Axioms limited to the form $P \circ Q \subset P$, are not sufficient for this application. For example, the "triangle" rule `isMAEBoundedBy(?x, ?y)` $\wedge$ `isMAEBoundedBy(?z, ?y)` $\wedge$ `MAE(?x)` $\wedge$ `MAE(?z)` $\wedge$ `GyriConnection(?y)` $\rightarrow$ `isMAEConnectedTo(?x, ?z)` cannot be represented in DL. An extension with some form of rules is required. Moreover, "ordinary" relations not defined in the ontology, also called "non DL" predicates are needed. They occur in rules, queries, or facts, e.g. the ternary predicate connects, or the binary predicate hasNoCommonPart etc. Ternary predicates are specially useful for representing the ground facts issued from the information extracted by the numerical tools, e.g. the initial fact `separates(s,m1,m2)` captures the separation relation between a sulcus segment `s` and two gyrus parts `m1` and `m2`, or `connectsMAE(op,m,g)` expresses the connection between three anatomical entities. Although it is possible to express a n-ary relationship with unary and binary predicates using reification, arbitrary arity is preferred. Hence, SWRL extension is not enough. OWL DL should be extended by a Datalog language supporting ontology concepts and roles in rule bodies or head as unary or binary predicates, and also non DL predicates, in particular n-ary predicates in body and head atoms, and negation in body atoms. [Gol05]

2.2 Fuzzy Identification of Brain Structures

2.2.1 Overview

Medical image processing and analysis is a highly emerging research area in bio-informatics technology. Let us here consider the applications of decision support in neurology and neurosurgery by processing of MRI (Magnetic Resonance Imaging) images. The goal of such applications is to assist the identification and lebeling of the various parts of an MRI image by using knowledge based techniques. The process usually involves two steps. In the first step the MRI image is automatically segmented into areas, each one associated with a set of features like the length and depth of a sulcus segment in a brain cortex, the connection of two sulcus segments, etc. The second step involves the identification and labelling of the different parts of the brain cortex, based on the segmented parts. Identification is the more difficult step. Such a process can be assisted by knowledge-based tools which provide both a component for representing domain knowledge in the form of ontologies as well as a rule component which captures the procedural aspect as well as dependencies between entities. For example, we could have the entities,

$$\begin{aligned} \mathsf{OPIFGyrus} &\sqsubseteq \exists \mathsf{isDAPartOf}.\mathsf{IFGyrus} \\ \mathsf{IFGyrus} &\sqsubseteq \exists \mathsf{isDAPartOf}.\mathsf{FrontalLobe} \end{aligned}$$

where $\sqsubseteq$ is a subsumption relation, $\mathsf{OPIFGyrus}$ represents the Orbital Pars of Interior Frontal Gyrus, $\mathsf{IFGyrus}$ the Inferior Frontal Gyrus [DGM04] and $\mathsf{isDAPartOf}$ represents the relation, $\mathsf{isDirectAnatomicalPartOf}$. Furthermore, using $\mathcal{SHIN}$ one can capture the facts that $\mathsf{isDAPartOf}$ is a sub-relation of a the broader relation $\mathsf{isAPartOf}$, that the relation $\mathsf{hasDAPart}$ is an inverse of $\mathsf{isDAPartOf}$, and that $\mathsf{isAPartOf}$ is a transitive relation. Following [GBDG05], we specify that $\mathsf{isDAPartOf}$ is an *inverse-functional* relation, writing $\leq 1\,\mathsf{Inv}(\mathsf{isDAPartOf})$ and meaning that there can be at-most one object that is a direct anatomical part of some other object. Now suppose that an image segmentation algorithm is applied to an MRI image in order to identify different brain parts. Since such algorithms cannot be sure about the membership or non-membership of an object to a certain concept, they usually provide confidence (truth) degrees. For example, we could have that o_1

$\mathsf{isDAPartOf}$ o_2 to a degree of 0.8, o_2 $\mathsf{isDAPartOf}$ o_3 to a degree of 0.9, o_2' $\mathsf{isDAPartOf}$ o_3 to a degree of 0.3, o_1 is an $\mathsf{OPIFGyrus}$ to a degree of 0.75, o_1 is a $\mathsf{IFGyrus}$ to a degree of 0.85 and that o_3 belongs to $\leq 1\,\mathsf{Inv}(\mathsf{isDAPartOf})$ to a degree of 0.7, meaning that it is likely that o_3 is connected only with one object. From that fuzzy knowledge one could deduce that the object o_3 belongs to $\exists\mathsf{hasAPart}.\mathsf{OPIFGyrus}$ to a degree of 0.75.

Moreover, the process of identification is assisted by a rule base which describes the dependencies between the relations of the brain cortex structures, as well as a brain ontology which stores the a priori expert knowledge about brain anatomy. For example there are rules of the form:

$$\mathsf{separatesMAE}(s, m_1, sm_2) \wedge \mathsf{hasAnatomicalPart}(m_2, sm_2) \wedge \mathsf{hasNoCommonParts}(m_1, m_2) \wedge \mathsf{SF}(s) \wedge \mathsf{MAE}(m_1) \wedge \mathsf{MAE}(m_2) \wedge \mathsf{MAE}(sm_2) \rightarrow \mathsf{separatesMAE}(s, m_1, m_2).$$

Obviously the degrees generated by the segmentation process should also appear in the application of the rules.

2.2.2 Limitations of existing languages

In [GBDG05], as well as here, it is noted that the results obtained by such a knowledge based application can be further refined if uncertainty reasoning is taken into consideration. This is because features like the depth or length of various parts of the brain cortex can vary between different individuals. This observation has led to approaches which also use uncertainty handling mathematical frameworks, like Fuzzy Set Theory [KY95], to perform the complex task of identification [GCWH+99].

2.3 Geospatial and Geographical Semantic Web Applications

2.3.1 Overview

Geospatial data is ubiquitous in many real-world applications. For example, it is useful for planning bus routes for a city, for finding suitable helicopter landing areas [CFB05], or in Location Based Services (LBSs), where Geographical Information Systems (GISs) are used to provide users with services relative to their current geographical position, like nearby restaurants or pharmacies and route planning or car navigation services.

In [CFB05] RDF/RDFS and OWL have been used to encode the semantics of geospatial data in the form of ontologies. For example concepts like “distanceFrom”, “busStopCandidate”, etc can be captured with the aid of an ontology language [CFB05]. Furthermore, in [CFB05] rules have been used for representing the semantics of the procedures for processing geospatial data. For example, the following rules can model an expert’s knowledge about candidate and non-candidate locations for bus stops:

$\mathsf{distanceFrom}(a, b, dist) \wedge \mathsf{lessThan}(dist, 100) \wedge \mathsf{isTypeOf}(a, RoadIntersection) \wedge \mathsf{isTypeOf}(b, ShoppingMall) \rightarrow \mathsf{busStopCandidate}(a)$,

$\mathsf{location}(a) \wedge \mathsf{existingBusStop}(b) \wedge \mathsf{distanceFrom}(a, b, dist) \wedge \mathsf{lessThan}(dist, 700) \rightarrow not(\mathsf{busStopCandidate}(a))$

where not represents negation-as-failure. The first rule expresses that a road intersection is a good bus stop candidate if it is less than 100 meters away from a shopping center. The second rule expresses that a location is not a good bus stop candidate if it is too close to an existing bus stop.

As is noted in [CFB05], the above approach can be enhanced if fuzzy reasoning is considered because many of the aforementioned predicates represent vague and imprecise knowledge. For example, the predicate lessThan, as it is implemented above, preforms a strict thresholding on the values that are below and above 100 meters. More precisely, a location which might have a distance of 97 meters is excluded from being a candidate bus stop location, even though it misses only for 3 meters to satisfy the restriction. By choosing to encode predicates like this with the aid of fuzzy concepts we can greatly gain in expressive power while avoiding the undesirable result we faced earlier. For example, one can replace the predicate lessThan with the fuzzy concept close. Now we can provide fuzzy values, ranging from 0 to 1, depending on how close a location is to the limit of 100 meters.

2.3.2 Limitations of existing languages

As we saw in the previous section, geospatial applications involve a number of vague concepts, like the concept “close”, “distant”, “nextTo”, and many more. Further refinements of applications of this type can be obtained by using different kinds of logical inference, such as default and fuzzy reasoning [CFB05]. Many of these techniques have been already used in GISs and other geographical applications [LGMR01].

2.4 Identification of Plant Species

2.4.1 Overview

Colours play an important role in the identification of plant species. A complete list of species containing those plants that have flowers of the requested colour can be very helpful to botanists in identifying a plant sample in nature. Colour descriptions of the same species are found in many different *floras*,[1] and are therefore treated as parallel sources. For instance, the species *Origanum vulgare* (Marjoram) has at least four colour descriptions of its flowers from four floras:

[1] A flora is a treatise describing the plants of a region or time.

- "violet–purple", in *Flora of the British Isles* [CTM87],
- "white or purplish–red", in *Flora Europaea* [THB+93],
- "purple–red to pale pink", in *Gray's Manual of Botany* [Gle63],
- "reddish–purple, rarely white", in *New Flora of the British Isles* [Sta97].

Some current projects [Act, RBG] attempt to store knowledge extracted from natural language documents in electronic forms. These projects generally allow keyword-based queries but do not support a formal representation of the semantics.

[WP05] present an ontology-based approach to tackle this issue. In order to decompose the semantics of colour descriptions, the authors propose a quantitative model based on the HSL (Hue Saturation Lightness) colour model. By using a parser based on a BNF syntax, they can quantify complex colour descriptions more precisely; for instance, they support adjective modifiers, ranges, conjunction or disjunction relations indicated by natural language constructions. Based on the semantics of colour descriptions, they can generate an ontology to model such complex colour information in our project. Such an ontology provides a foundation for information integration and domain-oriented query answering.

2.4.2 Limitations of existing languages

As shown in [WP05], user-defined datatypes are crucial to capture the semantics of the quantitative model. More specifically, customised datatypes are needed to represent the ranges for hue, saturation and lightness used in colour descriptions, as well as the degree adjectives in colour descriptions. This suggests that we cannot use the Semantic Web standard ontology language OWL DL for our purpose, since OWL DL does not support customised datatypes. The approach presented in [WP05] is applicable in other similar areas, such as the representation of leaf shapes, which is another key feature of identifying species. We have started to experiment with a quantitative model generated by a SuperShape formula [Gie03]. In this case, even user-defined datatypes are not enough, we will need to use user-defined datatype predicates to capture the numerical constraints in the SuperShape formula.

2.5 Semantic Web Services

Web Service technology has successfully abstracted the interface description of a component from concrete implementation details like the programming language used and the operating system with which a component is realized. Additionally, by using common Web protocols like HTTP the communication between entities over the internet has been made easier, such that integration on the syntactic level is now possible using Web Services standards.

However, most of the tasks related to the creation of distributed applications remain manual, such as the discovery of a particular kind of service, the composition of services to form bigger units, and the actual service execution. By adding semantic descriptions we envision increasing the level of automation. Semantic Web languages based on OWL, RDF(S), or service-tailored languages like WSML and extensions thereof shall serve as a basis here.

2.5.1 Overview

When focusing on discovery with syntactic descriptions one can compare the input and output signatures of a service to determine compatibility. However this is not enough for complete automation of service discovery, since (a) two identical signatures might not provide the same functionality and (b) two services with syntactically different signatures might provide the same functionality.

A service that does multiplication of two integers has the same I/O signature as a service that provides addition of two integers. We therefore require not only of a description of the input, but also a formalism that is able to express state changes, i.e. to express at least bi-state formulas that describe the relation between pre-service and post-service state. Thus, services with the same signature might provide different capabilities.

On the other hand, consider two weather services that provide the temperature in a given city in Austria: both accept an integer value representing the ZIP code as input and provide the temperature in Celsius as output, however one requires the date for which the temperature shall be returned as additional input whereas the other returns the current temperature. Given background knowledge of the current date both services (despite different signatures) are able to provide the current temperature. Thus services with different signatures might under certain circumstances serve the same goal.

Also, the I/O signature needs to be described semantically. We need languages which allow description of the required inputs of a service in terms of ontologies and constraints on the input given to that service. As we will see in the next subsection, current ontology languages like RDF(S) and OWL have particular limitations here.

2.5.2 Limitations of existing languages

We briefly give examples of two limitations of current Semantic Web languages in the context of semantic web service description.

Input Checking Current ontology languages like OWL and RDF are only applicable to a limited extent for checking the validity of closed sets of information. This is mainly due to the fact that there is no way to apply the closed world assumption. We briefly discuss certain problems in this context.

Let us assume that the inputs and outputs of a Web service are described using concepts from an ontology. One use of these descriptions is to check if the instance data in a particular knowledge base of an agent is suitable as input for that Web Service.

Concept descriptions using OWL imply the use of the Open World Assumption (OWA), which might be problematic in our case: Let us assume a class *SearchRequest* describes (from the provider perspective) a web service which allows searching for books based on keywords:

Class(SearchRequest partial hasKeyWords minCardinality(1))

The closed knowledge base of an agent trying to communicate with this service might contain only the assertion *Individual(myRequest type(SearchRequest))* along with some other values such as the intended author, but this knowledge base might not define any keywords. Intuitively we would expect that we cannot provide the required input. However, naïve application of OWL reasoning to a simple union of the knowledgebase and the service description will infer the existence of at least one *keyword* [dBPLF05]. Making use of OWL reasoning to determine whether suitable inputs are available provides a representational challenge.

Functional Description Describing the inputs and outputs using concepts from an ontology improves Web service related tasks such as the discovery process. However different services having identical input and output concepts cannot be distinguished. E.g. a service that increases the balance of an account and one that decreases the balance of an account cannot be distinguished when the output is only the new balance.

In order to describe precisely the relation between inputs and outputs the language used must support bi-state formulae: rules describing the effects of a service with respect to certain input and the pre-state of the service execution. In principle, an extension of a language like SWRL could be used, however at present these rule languages only express mono-state sentences and in OWL-S, for example, there is no clean definition of how to define relations between states before and after execution.

2.6 SemVersion: Versioning RDF and Ontologies

2.6.1 Overview

The overall goal is a standard versioning system or methodology that can provide a common way to handle versioning issues. SemVersion provides structural (purely triple based) and semantic (ontology language based, like RDFS) versioning. It separates language-neutral features for data management from language-specific features like semantic diffs in design and implementation. This way SemVersion offers a common approach for already widely used RDF models and a wide range of ontology languages. The approach is inspired by the classical CVS system for version management of textual documents (e.g. Java code). A core element of the approach is the separation of language-specific features (the semantic diff) from general features (such as structural diff, branch and merge, management of projects and metadata). A speciality of RDF is the usage of so-called blank nodes. As part of the approach a method is presented for blank node enrichment using inverse functional properties which helps in versioning of blank nodes.

SemVersion requires two main features which are not currently included in the Semantic Web languages:

- context information
- bi-temporal database features

As context information the interest is in data provenance, and links to external information such as temporal and user-defined versioning information.

With regard to temporal information the interest is in defining data types and operators related to bi-temporal databases. Several types have been defined in the JAVA prototype such as time interval, time expression, valid time, transaction time, UC, now, and so on—typical data types and constants found in bi-temporal databases. Both point-based and interval-based operators have been or are currently being implemented.

Query languages are also a concern. From the entire set of Semantic Web query languages, two main groups can be differentiated: the SQL-like query languages and the DL-based query languages. SQL-like query languages include research, de facto standards and W3C standards such as RQL, RDQL, SerQL, N3QL and SPARQL. The DL-based group includes OWL-QL. SQL-like query languages are much more intuitive for people experienced in the field of databases and SQL programming whereas DL-based query languages are more suitable for people experienced in logic programming. It would be interesting to provide translators from DL-based query languages to SQL-like query languages and vice versa. Also it would be interesting to include explain modules and features for DL-based queries. In other words SQL-like query languages are more suitable for human users whereas DL-based query languages such as OWL-QL are more suitable for agent-to-agent conversations as indicated in [PFT$^+$04b].

2.6.2 Limitations of existing languages

This use case directly addresses one major perceived limitation of semantic languages—difficulty in managing versions—through creation of a comprehensive external architecture. The approach pushes the boundaries of OWL's "annotation" facilities for context information. Furthermore, the rich datatypes used in bi-temporal databases stress the value of more robust datatype support.

A focus on query languages also highlights a major limitation which has so far escaped mention: no consensus on semantic query languages yet exists.

2.7 Information Integration

Two very simple use cases are presented which are believed to be prototypical of integration among several ontologies.

2.7.1 Overview

Overlapping terminology

Two data sources on the Web represent similar notions. An ontology states that source A describes Books (among other things) and source B describes Articles. We wish to formulate a rule which imports all books in A as articles in B.

Unknown information

We wish to define a concept "interesting movie" containing those instances described by some of the available (web) resources as movies, but not those made by director "Ed Wood". We are well aware that our information sources are incomplete, and while we require that at least one source provides information on whether an instance is a movie or not, mere lack of available information that Ed Wood was involved is enough to classify an instance as interesting.

2.7.2 Limitations of existing languages

Given two ontologies consisting of structural axioms (TBox) and assertions about data (ABox), OWL provides only limited means to combine these ontologies, namely the complete import of one ontology into the other by the $import$ construct. Wholesale merging of the two data sources is not the usage scenario described above (which demands import only of those instances known to be Books). Extensions like C-OWL [BGvH$^+$03] address this issue by defining a requirement for more fine-grained bridge rules between ontologies.

The "interesting movie" class cannot be expressed in OWL because OWL does not include a construct describing instances which are not necessarily known to be members of another class. This feature is often provided as “negation as failure” in logic programming environments.

2.8 Speaking the language of business experts

2.8.1 Overview

In many domains the ontology building process is difficult and time consuming. From practical experience, this is not because these domains are not well understood or a consensus cannot be found, but typically because it is difficult for domain experts to understand and use ontology languages. Indeed, current ontology languages and tools still require an understanding of their underpinning logics. Ontologies are currently only being built by knowledgeable people with proficiency the art of ontology engineering as well as extensive domain knowledge; or, by ontology engineers who elicit some knowledge by interviewing experts. Our motivation is to enable business domain experts to build ontologies starting from the language they speak, and the “logic” they think of. This is not only to decrease ontology development costs, but also to increase quality of ontological content as validated by the domain experts themselves.

2.8.2 Limitations of existing languages

The limitation of ontology languages that we present here is not in the sense of lack of expressiveness or logical operators but the capability (or lack thereof) of these languages to be used by subject matter experts. Consider as an analogy the differences between the cost and the skills needed to program in Assembly versus C++ versus Visual Basic. Notice also that these languages are built on top of each other, i.e. C++ using Assembly and Visual Basic using C++. The inability of business users to work with OWL and RDF should be considered a limitation in its own right.

2.9 Classification of Protein Data

2.9.1 Overview

The volume of genomic data is increasing at a seemingly exponential rate. In particular, *high throughput* technology has enabled the generation of large quantities of DNA sequence information. This sequence data, however, needs further analysis before it is useful to most biologists. This process, called *annotation*, augments the raw DNA sequence, and its derived protein sequence, with significant quantities of additional information describing its biological context.

One important process during annotation is the classification of proteins into different families. This is an important step in understanding the molecular biology of an organism. Attempts to automate this procedure have, however, not generally matched the gold-standard set by human experts. Human expert classification has been more accurate because their expertise allows them to recognise the properties that are sufficient, for example, to place an individual protein into a specific subfamily class. Automated methods have, in contrast, often failed to achieve the same level of specificity. A goal, therefore, has been to improve the precision of automatic protein classification, and bring it up to the same level as that achieved by human experts.

Given a set of proteins, each with a (partial) description of its properties, the objective is to find, for each of these proteins, the most specific protein family classes of which it is an instance. To describe protein family classes, an OWL-DL ontology is used; this enables the specification of necessary and sufficient conditions for a protein to be an instance of a given protein class.

The ontology models the biology community's view of the current knowledge of protein classification. The protein data is derived using standard bioinformatics analysis tools, these data are translated into OWL-DL instance descriptions that use terms from the ontology, and automated reasoners are used to classify these instances. [WBH+05]

Knowledge Modelling

One protein family, the *protein phosphatase* family, was used as a case study to demonstrate a new, ontology-based method for automated annotation. This method was designed to combine the speed of automated annotation with some of the detailed knowledge that experts use in annotation.

In general, proteins are relatively modular and comprise a number of different *protein domains*. Using a protein sequence, it is often possible to computationally determine the protein domains of which it is composed. For many protein families, including the protein phosphatases, it is possible to classify their members based on the protein domains of which they are composed. To avoid confusion with interpretation domains or the domain of a property, we use "p-domain" for protein domain.

The different p-domain compositions of proteins suggests the specific function of a protein. Individual p-domains, however, often have specific and separate functions from the protein as a whole. For example, an enzyme will have a catalytic p-domain that performs the catalysis on the substrate molecule, but it will also contain structural p-domains and binding p-domains that ensure that the substrate can interact with the catalytic p-domain. Therefore, a specific combination of p-domains is required for a protein to function correctly. In some cases, the presence of a certain p-domain is *diagnostic* for membership in a particular protein family, i.e., some p-domains only occur in a single protein family. If a protein contains one of these diagnostic p-domains, it must belong to that particular family. For example, the protein tyrosine kinase catalytic p-domain is diagnostic for the tyrosine kinases.

Most protein families are, however, defined by a non-trivial combination of p-domains. For example, as you descend the hierarchical structure, extra p-domains (and therefore more specific functional properties) are observed in the protein class definitions. For example, an R5 phosphatase is a type of classical receptor tyrosine phosphatase. As a tyrosine phosphatase, it contains at least one phosphatase catalytic p-domain and, as a receptor tyrosine phosphatase, it contains a transmembrane region. The R5 type actually contains two catalytic p-domains and a fibronectin p-domain, identifying it as an instance of even more specific subclasses.

Identifying the p-domain composition of a protein is, therefore, a first step towards its classification. There are databases describing functional p-domains, for example, PROSITE [HSLS+04], SMART [LCS+04] and INTERPRO [MAA+05], and these databases come with specific tools, such as INTERPROSCAN, which can report the presence of these p-domains in a novel protein sequence. Bioinformaticians are, however, usually required to perform the analysis that places a protein (with its set of p-domains) into a particular protein family.

The method presented could be applicable in general to many protein families, but to demonstrate the technique and the fine-grained classification possible, the analysis of one family is described: the protein phosphatases, in the human and *Aspergillus fumigatus* genomes.

Building the ontology

All the information used for developing the ontology comes from peer-reviewed literature from protein phosphatase experts. The family of human protein phosphatases has been well characterised experimentally, and detailed reviews of the classification and family composition are available [ASB+04, Coh97, Ken01]. These reviews represent the current community knowledge of the relevant biology. If, in the future, new subfamilies are discovered, the ontology can easily be changed to reflect these changes in knowledge.

Fortunately for this application, there are precise rules,[2] based on p-domain composition, for protein family membership, and these rules can be expressed as class definitions in an OWL-DL ontology. The use of an ontology to capture the understanding of p-domain composition enables the automation of the final analysis step which had previously required human intervention, thus allowing for full automation of the complete process.

The ontology was developed in OWL-DL using the Protégé editor,[3] and currently contains 80 classes and 39 properties; it is available at `http://www.bioinf.man.ac.uk/phosphabase/download`. Part of the subsumption hierarchy inferred from these de-

[2] We use "rules" here in a completely informal way.

[3] Protégé 3.0 with OWL plugin 1.3, build 225.1.

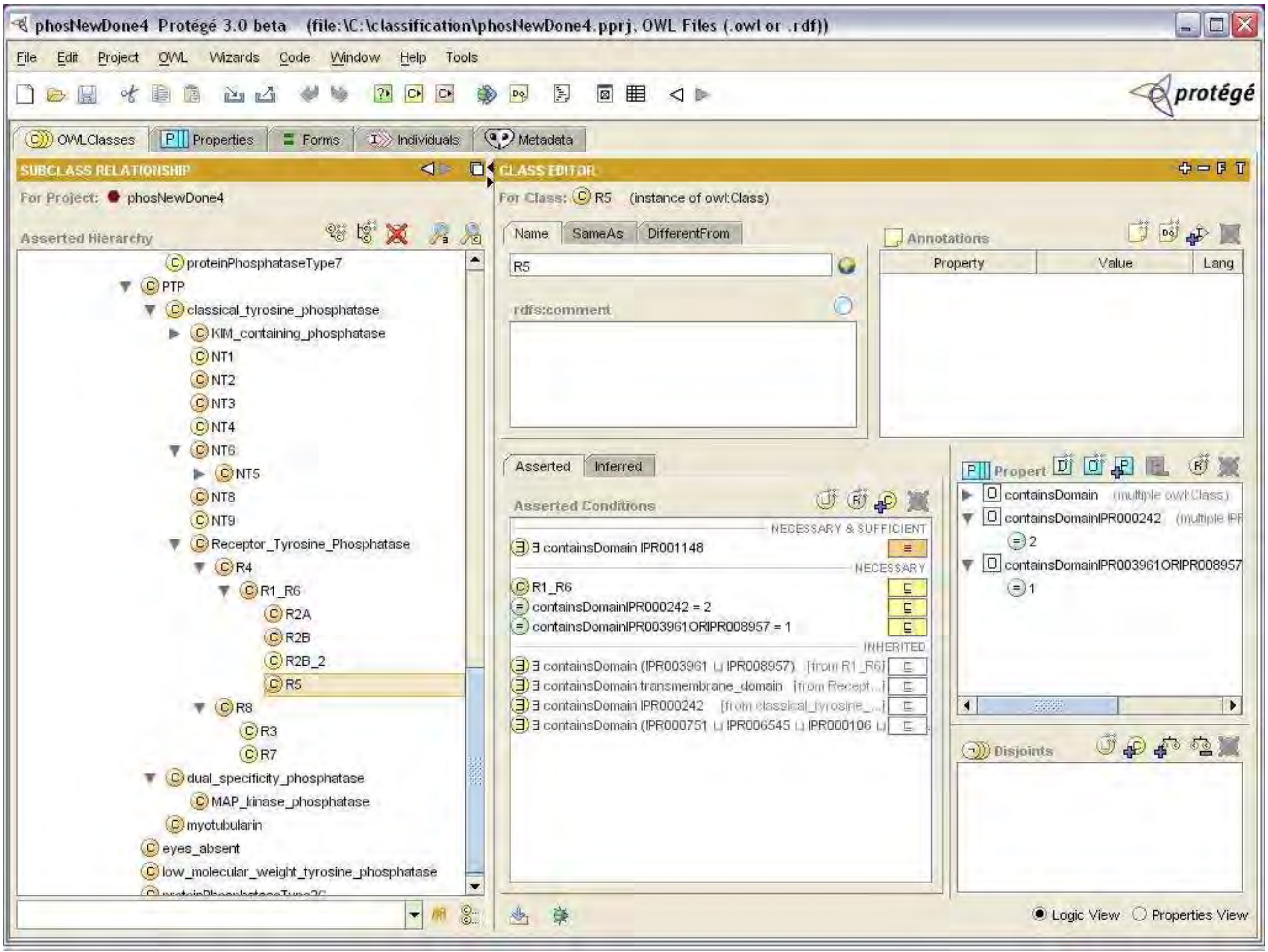

Figure 2.1: A screenshot of the phosphatase ontology in the OWL ontology editor Protégé.

scriptions can be seen in the left-hand panel of Figure 2.1, which shows the OWL ontology in the Protégé editor.

More precisely, for each class of phosphatase, this ontology contains a (necessary and sufficient) definition. For this family of proteins, this definition is, in most cases, a conjunction of p-domain compositions:

If a Y protein contains at least n_1 p-domains of type X_1 and …and at least n_m p-domains of type X_m, then this protein also belongs to class Z.

For example, receptor tyrosine phosphatases contain one or two phosphatase catalytic p-domains, and receptor tyrosine R2B phosphatases contain exactly 2 tyrosine phosphatase catalytic p-domains, one transmembrane p-domain, at least one fibronectin p-domain, and at least one immunoglobulin p-domain.

In some cases, X_i is a disjunction of p-domains. P-domains come with a rather "flat" structure, i.e., only few p-domains are specialisations of others.

Analyzing results

For the human phosphatases, automated protein classifications were compared with the thorough human expert classification. The class hierarchy of the phosphatase ontology was browsed and, for each class, those proteins for which the system inferred that this class was the most specific one were retrieved. Instances that did not fit any of the ontology class definitions (i.e., whose most specific class was the top class) were also identified.

For the *A.fumigatus* phosphatases, the class hierarchy was browsed in a similar way but, as the phosphatases from this organism were less well characterised, the differences between the human and *A.fumigatus* set were of particular interest, i.e., classes were identified which had instances of the human proteins, but not of the *A.fumigatus* proteins, and vice versa.

2.9.2 Limitations of existing languages

The "counting" statements used in expressions such as:

If a Y protein contains at least n_1 p-domains of type X_1 and ... and at least n_m p-domains of type X_m, then this protein also belongs to class Z.

are not a part of the OWL language, but are absolutely essential to accurate modelling of domain knowledge. A workaround was applied: for each X_i that would been used in a qualified number restriction, a subproperty `containsX_i` of `contains` was introduced, and the range of `containsX_i` was set to the class X_i. In addition, sub-property assertions were added so that the hierarchy of newly introduced properties `containsX_i` reflected the class hierarchy of the classes X_i. This workaround is not always equivalent to true qualified number restrictions, however, which was a surprise to the ontology engineers involved. Additional analysis was required to prove correctness in the current context, and it is unrealistic to expect similar future projects to have access to the expertise necessary to conduct similar analyses.

2.10 Multiple Viewpoint Representation for Oncology

Oncology is a complex domain where several specialties, e.g. chemotherapy, surgery, and radiotherapy are involved in a sequence of treatment phases. In most cases, the adequate therapeutic decision is taken according to a protocol that associates standard patient characteristics with a recommended treatment. The protocol is structured as a kind of decision tree with respect to the oncology specialties, meaning that a specialty in oncology organizes domain knowledge and past experiences in a specific way. Although designed to take into account the majority of the medical cases, the protocol does not cover all the situations. Decisions involving patients out of the protocol are elaborated within a multi-disciplinary expert committee meeting, and rely on the adaptation of the solutions provided by the protocol for similar cases. During an expert committee meeting, an expert from each specialty provides a personal view on the treatment recommendation as a part of a collective recommendation. For each specialty, different characteristics of the patient are analyzed and taken into account for setting on a specific treatment within the whole treatment process. In this way, the oncology specialties provide different interrelated viewpoints for a patient treatment, i.e. information on a given patient in a local viewpoint can be shared with other information in another local viewpoint to build a global treatment. Moreover, a decision taken in a local viewpoint, i.e. for a particular oncology specialty, may have an influence on the decision to be taken in another local viewpoint.

A protocol contains the standard knowledge for decision support in oncology. As a standard Web formalism for knowledge representation and exchange, OWL is a well-suited language for the formalization of the knowledge contained in a protocol. Furthermore, reasoning mechanisms associated with OWL, such as classification and instantiation, may be used for decision support in oncology. However, knowledge representation and reasoning have to consider (and take advantage of) the multiple viewpoints involved in the decision, corresponding to the oncology specialties.

2.10.1 Overview

Oncology, like many other medical domains, involves several medical disciplines. Thus, decision knowledge included in the protocols combines aspects based on the different disciplines, i.e. several viewpoints. Each of these viewpoints constitutes a suitable way within a discipline for representing, organizing and using decision knowledge. Indeed, a cancer specialist will use different elements of knowledge, and different representations of the patient according to his interest, e.g. for establishing a surgical treatment or a chemotherapy. These multiple viewpoints are visible in the structure of the protocols. For example, the protocol for breast cancer treatment is based on several "branches", each of them focusing on a particular phase of the medical treatment, e.g. preoperative chemotherapy, surgery, radiotherapy, and complementary treatment, involving one or more disciplines of oncology, among chemotherapy, surgery, radiotherapy and hormonotherapy. A branch has its own vocabulary, and is interested in providing a particular recommendation, in a given phase of the treatment, using only specific characteristics for this task. In addition, the expert committee whose role is to adapt the content of the protocols for cases out of the protocol, i.e. cases for which the direct application of the protocol is not possible, gather experts of the different oncology disciplines. Each expert brings his own contribution to the solution, according to the discipline he depends on, and collaborates

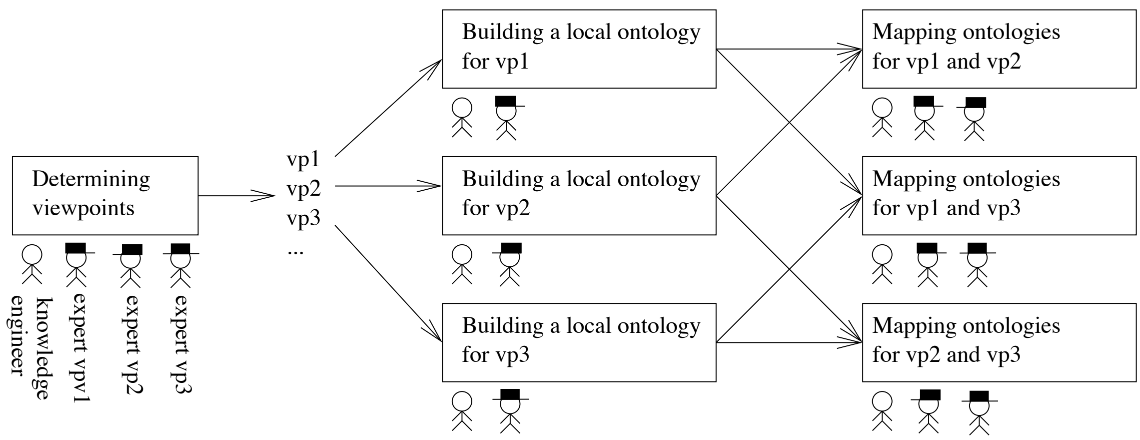

Figure 2.2: An approach for representing knowledge according to multiple viewpoints.

with the experts from other disciplines, in the construction of a global solution.

Modeling Viewpoints

Usually, the construction of an ontology is based on a consensus between actors of the domain, in order to formalize the knowledge units shared by the actors, and to represent these units independently of any context of use [Gua97]. Here, the idea is different and much more related to the practise. The ontology construction takes into account the viewpoint of each actor. Within this framework, a viewpoint corresponds to the representation of the knowledge useful for a given application, or a given group of persons, collaborating with another one. The framework should enable to represent viewpoints within contexts or "local ontologies". Mappings between contexts allow the exchange and the reuse of knowledge units across contexts, thus allowing the combination of local viewpoints in a global representation.

The approach to model and to formalize multiple viewpoints is distributed, or decentralized, i.e. there is no need to set up a consensus, but on the contrary to distinguish viewpoints. Three steps are the necessary (see figure 2.2):

1) Determining viewpoints. This step consists in determining the viewpoints to be represented. A domain that is vast and complex is often organized according to several services, tasks, working groups or communities. This organization provides an *a priori* division of the domain within viewpoints, that can be reused in this step.

2) Building a local ontology. Here, the objective is to build a local ontology corresponding to a viewpoint. A local ontology includes the knowledge considered to be useful within the current viewpoint, *independently* from the other viewpoints. This means that the local ontology only implements relevant knowledge units in the viewpoint.

3) Mapping ontologies. The relations existing between the viewpoints are materialized through mappings between local ontologies. Mappings should be defined between the ontologies.

We now describe in more detail how to define a view point.

Determining viewpoints

Determining the relevant viewpoints is a difficult task, and consists in finding elementary segments of the domain knowledge, independent from each others, but interrelated (i.e. a collaboration is possible). The protocol for breast cancer treatment can be considered as composed of four main treatment phases involving the oncology disciplines:

1. The presurgical chemotherapy is a chemotherapy treatment that may be recommended before surgery, and main objective is to slow down, and even stop the progression of the tumor, for allowing a lighter surgery.

2. The surgery is the central phase of the treatment. The objective of this segment is to decide the surgical treatment that must be performed, according to the characteristic of the patient.

3. The radiotherapy aims at supplementing surgery by eliminating the still present and possible infected cells, in particular in the lymph nodes. The objective is to decide the parts of the anatomy of the patient to be irradiated.

4. The complementary treatment may be a chemotherapy or an hormonotherapy, aimed at completing surgery.

These segments and decision elements are interrelated, although they may exist on their own, i.e. in some cases, the decision taken in a Mappings should be directional: a bridge rule is a knowledge unit defined within a particular context, and thus is only usable within the corresponding viewpoint. An important question concerns the knowledge units that can be “seen” from a given viewpoint, and that can be “reused” within a viewpoint, in order to make precise the correspondences between viewpoints, i.e. how a given context may take advantage of knowledge form another context. segment may depend on the decision taken in another segment. This segmentation of the knowledge is daily used by physicians, and seems to be well-suited to their task. Thus it is natural to rely on such a segmentation to set up the viewpoints. Another choice could be to consider two different viewpoints for the complementary treatment segment of the protocol, one for chemotherapy and the other for hormonotherapy, or to group presurgical chemotherapy and complementary chemotherapy. But it is preferable to preserve the initial structure of the protocol. In the following, we rely on four main viewpoints, corresponding to presurgical chemotherapy, surgery, radiotherapy and complementary treatment, represented within four local ontologies, namely O_{pc}, O_s, O_r and O_{ct}.

Building local ontologies

The viewpoints considered here are related to the recommendation of treatments for patients ill with breast cancer. The classes of patients (`Patient`) and treatments (`Treatment`) are present in the four contexts. Nevertheless, these classes do not represent necessarily the same set of elements in different viewpoints, i.e. they should not be interpreted in the same way according to the context. For example, a patient in O_s is a person having a breast cancer, whereas a patient in O_{tc} is a woman[4] having a breast cancer, and for whom a surgical treatment has already been proposed. The link between the patients and their recommended treatments is represented in all the viewpoints by a property named `recommendation`.

Presurgical chemotherapy corresponds to a set of cures of several drugs (called FEC 100), whose results are evaluated at the end of the third cure [(RS05a]. The decision to be taken has an implication on the continuation of the cures, according to the evolution of the tumor. If the tumor regresses or is stabilized, chemotherapy must be continued, otherwise the medial case may be treated during an expert committee meeting. The response of the chemotherapy should also be evaluated, to check whether it is sufficient to allow a "preserving surgery". In the domain of breast cancer treatment, a preserving surgery is a partial ablation which preserves the essential parts of the breast.

For example, we consider the following part of the O_r context:

$$\begin{aligned}\texttt{P-RepSR} \equiv \texttt{Patient} \sqcap \exists \texttt{PSChemo}.(&\texttt{PreSurgicalChemo} \sqcap \\ &\exists \texttt{response}.(\texttt{Regression} \sqcup \texttt{Stabilization})\end{aligned}$$
$$\texttt{P-RepSR} \sqsubseteq \exists \texttt{recommendation}.\texttt{ChemoContinuation}$$

This is indicating that the protocol recommends to continue the chemotherapy (`ChemoContinuation`) in the event of stabilization or of regression of the tumor ($\exists\texttt{response}.(\texttt{Regression} \sqcup \texttt{Stabilization}$).

The surgery is the central phase in the treatment of a patient ill with breast cancer. To simplify, we will only consider here the decision knowledge related to the surgery of a particular kind of tumors, for which a direct ablation is possible [(RS05b]. A treatment in this phase can be a partial ablation of the breast, or a total ablation (i.e. a "Patey ablation"). A partial ablation corresponds to a preserving ablation of the breast, concentrating only on

[4]Some hormone based treatments cannot be applied on men.

the tumor. It is associated with a "curettage" (said "of the two lower steps"). A curettage corresponds to the extraction of lymph nodes for being analyzed. The mention "of the two lower steps" is related to the location where the lymph nodes have to be extracted. A total ablation, or a Patey ablation, is associated with a "complete curettage", said to be "of the three steps". For example, a patient with a directly operable and not a multifocal tumor, may be be treated with a partial ablation of the breast. This may be represented by the following axioms in O_s:

$$
\begin{aligned}
&\texttt{P-DOT-NMF} \equiv \texttt{Patient} \sqcap \\
&\qquad \exists \texttt{hasTumor}.(\texttt{DirectlyOperableTumor} \sqcap \exists \texttt{hasFocus}.\texttt{SimpleFocus}) \\
&\texttt{P-DOT-NMF} \sqsubseteq \exists \texttt{recommendation}.\texttt{PartialAblation}
\end{aligned}
$$

The radiotherapy aims at supplementing the surgery by eliminating the possibly remaining infected cells, in particular in the lymph nodes. It consists in choosing the zones to be irradiated, among the breast, the frontier of an ablation zone, the zone under the clavicle, or the internal mammary chain (IMC), possibly extended (EIMC) [(RS05c]. The characteristics of the patient taken into account for the decision are related to the recommended surgery, the localization of the tumor in the breast, its size and the presence of infected cells in the lymph nodes.

$$
\begin{aligned}
&\texttt{P-PS-NILN-TII} \equiv \texttt{Patient} \sqcap \exists \texttt{surgery}.\texttt{PreservatingAblation} \sqcap \\
&\qquad \exists \texttt{infectedLymphNode}.\{false\} \sqcap \\
&\qquad \exists \texttt{hasTumor}.(\exists \texttt{localization}.\texttt{Infero-Internal}) \\
&\texttt{RadioBreastEIMC} \equiv \texttt{Irradiation} \sqcap \exists \texttt{zone}.\texttt{Breast} \sqcap \exists \texttt{zone}.\texttt{EIMC} \\
&\texttt{R-BreastEIMC} \equiv \exists \texttt{recommandation}.\texttt{RadioBreastEIMC} \\
&\texttt{P-PS-NILN-TII} \sqsubseteq \texttt{R-BreastEIMC}
\end{aligned}
$$

The axioms above indicate that a patient with a preserving surgery recommendation, lymph nodes that are not infected, and a tumor located in the infero-internal part of the breast, must be treated by a radiotherapy of the breast, and of the extended internal mammary chain. It should be noticed that the recommendation *no radiotherapy* is allowed, using an axiom such as follows, stating that for patients represented by instances of the class `P`, a radiotherapy is not recommended.

$$\texttt{P} \sqsubseteq \neg \exists \texttt{recommendation}.\texttt{Radiotherapy}$$

The largest part of the protocol for breast cancer treatment concerns the complementary medical treatment [(RS05d]. It focuses on the recommendation of an hormonotherapy or a chemotherapy for complementing the surgery. The hormonotherapy is related with the control of the hormones having an influence on the development of the tumor. There are two types of hormonotherapy used for breast cancer treatment. The suppressive hormonotherapy consists in the suppression of the functions of the ovaries, producing hormones, in a chemical or a surgical way (ablation of the ovaries). The additive hormonotherapy consists in a cure of Tamoxifene, an anti-estrogen product. Usually, chemotherapy consists of 4 to 6 cures of injections of drugs called FEC 50 (chemotherapy of level 1) or FEC 100 (chemotherapy of level 2). The choice of the complementary treatment is guided by characteristics such as the status of the hormonal receptor, the number of invaded lymph nodes, the age of the patient, as well as the size of the tumor and its grade (1, 2 or 3). For example, the protocol states that a patient whose age is less than 35, having no invaded lymph nodes, positive hormonal receptors, and a tumor of grade equal to or higher than 2, or of size equal to or higher than 1 cm, has to be treated by a chemotherapy of level 1:

$$
\begin{aligned}
\texttt{P-ALE35-NILN-HRP-TGS2orTSS1} \equiv\ & \texttt{Patient} \sqcap \\
& \exists \texttt{age}.\texttt{infe35} \sqcap \exists \texttt{NBInfectedLymphNodes}.\{0\} \sqcap \\
& \exists \texttt{hormoneReceptor}.\texttt{Positive} \sqcap \\
& \exists \texttt{tumor}.(\exists \texttt{grade}.(\texttt{grade2} \sqcup \texttt{grade3}) \sqcup \exists \texttt{size}.\texttt{supe1}) \\
\texttt{P-ALE35-NILN-HRP-TGS2orTSS1} \sqsubseteq\ & \exists \texttt{recommendation}.\texttt{Level1Chemo}
\end{aligned}
$$

where `supe35` and `supe1` are datatypes, respectively representing integers equal to or lower than 35 and floats equal to or higher than 1.

Mappings

This last step consists in representing the relations between the four viewpoints described above by so-called mappings. Each local ontology stands for the vision of one or two disciplines of oncology, involved in a particular phase of the treatment. As surgery takes place after presurgical chemotherapy, it seems natural that the decisions in surgery follow the decisions in presurgical chemotherapy. In the same way, radiotherapy and complementary treatment viewpoints depend on surgery, and then may take advantage of knowledge units lying in the surgery viewpoint, in order that a decision may be taken and applied in these two viewpoints. The figure 2.3 summarizes relevant mappings according to the principles introduced above. An arrow from a context O_i to a context O_j indicates a non empty mapping M_{ij}.

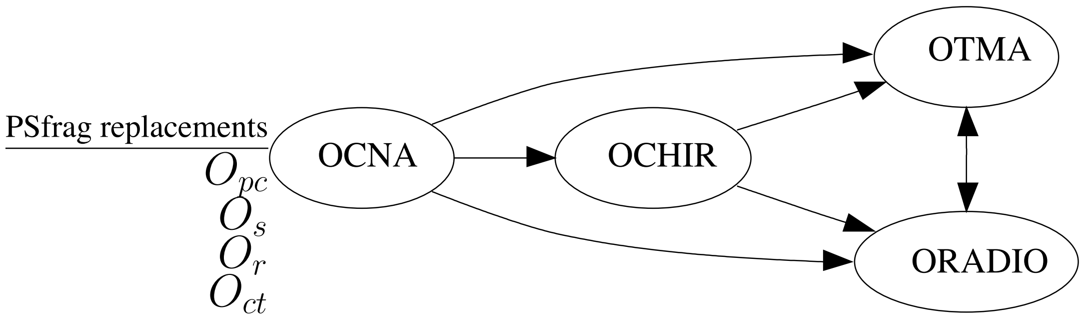

Figure 2.3: Mappings between viewpoints (contexts) for the representation of the protocol for breast cancer treatment.

Advantages of a Multiple Viewpoint Representation

The first element to be considered for evaluating the strength of the viewpoint representation may be the expressivity allowed by a distributed representation language. However, the relevance of this element may be discussed, according to the fact that, formally, the distributed representation language may not be more expressive than standard OWL, as shown in [BS02]: any knowledge base represented within a distributed description logic can be transformed without loss of information into a knowledge base in the standard $\mathcal{SHIQ}$ description logic. Thus, a set of local ontologies can be grouped in a standard ontology. Nevertheless, [BGvH$^+$04] and [ST05] introduce the principle of *non propagation of the inconsistency*, that is not taken into account in the transformation described in [BS02]. This principle states that the inconsistency of a local ontology does not have any influence on the coherence of the other local ontologies of the ontology space. Actually, one goal when building the viewpoints in the application is to provide coherent viewpoints, locally as globally.

The representation of viewpoints has an obvious interest in terms of knowledge engineering and knowledge management. The decentralized approach, i.e. based on viewpoints or local ontologies represented by contexts, consists in building a set of local ontologies, each corresponding either to a part of the domain, or to a given task, a given application or a group of interested persons. These ontologies are then mapped to each other by the mean of bridge rules. In this way, it is no more necessary to set up a global consensus on the domain knowledge. A viewpoint materializes a (more) simple and homogeneous representation of the knowledge considered to be useful according to a given interest. Moreover, the construction and the maintenance of a set of contexts appear to be simpler than the building and the maintenance of a unique large ontology aggregating the knowledge distributed in all the viewpoints. The evolution of the knowledge represented in a particular viewpoint may be realized locally, without need to make references to the global representation. Furthermore, the viewpoint representation simplifies the use and the access to knowledge units. The actors of the domain, according to a given viewpoint, may concentrate only on the relevant contexts, built to be well-suited to their uses.

The collaboration and knowledge exchange between viewpoints are then carried out and controlled automatically, through mappings.

For example, regarding the above application for breast cancer treatment, a previous representation of the protocol does not include any viewpoints representation [dBLN04]. The building of this global and unique representation is supposed to integrate the whole domain knowledge units, coming from the different disciplines. However, representation choices are necessary for certain elements, that are not always in accordance with the practice of the domain actors. In particular, because of the needs related to the complementary treatment, it has been decided to replace the representation of the status of lymph nodes with the number of invaded lymph nodes, leading to a disagreement with the usual representation of the lymph node status in radiotherapy. Moreover, any emergence of a new technique, e.g. in surgery, leads to update knowledge units among more than the thousand classes constituting the unique representation of the protocol. A single evolution of the protocol may potentially be the cause of large modifications and updating of the representation, including elements possibly without any direct relation with the updated knowledge units. The inclusion of viewpoints in the representation of the protocol allows a more simple knowledge acquisition and representation, knowing that a viewpoint is made of between ten to twenty classes only, evolving independently form each other inside the viewpoints.

2.10.2 Limitations of existing languages

This modelling pattern directly highlights the extremely limited ability to "modularize" OWL ontologies, and the problems this limitation causes for large-scale projects and deployments which involve a number of different actors.

Reasoner performance on such tasks as subsumption and instantiation are also significant limitations to such large-scale projects. Whether this is considered a limitation of the OWL language or of current tools, such issues cannot be dismissed.

In fact, the above two issues converge when one considers the potential for decentralized reasoning over decentralized ontologies: individual actors are expected to focus on their own viewpoint and their own subsumption lattice, thus the ability to restrict reasoning and hierarchy computation to a particular viewpoint could prove valuable.

The ability to restrict processing to a particular viewpoint could also be considered a crucial element of any semantic query language; lack of such a feature from any potential consensus language would be a severe limitation.

Chapter 3

Requirements

In this chapter, approaches to overcoming some of the limitations identfied in Chapter 2 are described. While most problems can be overcome with substantial changes to architecture (e.g. using procedural logic in a traditional programming language to achieve the desired results), we focus on approaches which continue to represent knowledge in semantic languages. The extension of OWL and RDF with additional features for increased expressiveness or usability is considered (qualified number restrictions; uncertainty handling; rich datatypes), as well as the use of additional or alternative semantic layers (closed-world assumption; logic programming). Pure "usability" enhancements are also considered, which could make semantic languages easier to work with in common architectures, even when their formal power remains unchanged.

The potential impact of each approach is summarized with respect to the use cases described in Chapter 2 as well as those identified by industrial work packages in deliverable 1.1.4 [LN05] and other commonly-encountered usage patterns. Because there are a number of ways that each approach could be followed to fulfill a given requirement, the cited use cases can be considered partial benchmarks against which success of an implementation of the given proposal can be evaluated.

3.1 Qualified number restrictions

3.1.1 Definition of the requirement

It is often useful to express constraints such as "has exactly four parts that are legs", "has at least two groups that are phosphate groups", "has exactly one feature that is temperature", etc. In each of these cases, we want to constrain not the total number of values for a property, but rather the number of values of a given type. Such restrictions are called "qualified cardinality restrictions" (QCRs) or "qualified number restrictions" (QNRs) because they are "qualified" by the type of the value. They are supported in most modern description logics but were omitted from the final version of the OWL standard.

OWL currently provides unqualified number restrictions via the `maxCardinality`, `minCardinality`, and `cardinality` class constructors. For example, we can state that a “Minimal Italian Dinner” contains exactly three courses:

```
Class(Minimal_Italian_Dinner,
    subClassOf( Restriction( has_course, cardinality(3) )))
```

Further, qualified restrictions with minimum cardinality 1 are equivalent to OWL `someValuesFrom` class constructors.

3.1.2 Background work

Qualified number restrictions are well-understood constructs. For all common Description Logics which already include unqualified number restrictions (including that on which OWL is based), the worst-case complexity of reasoning remains the same when they are extended with qualified number restrictions ([Tob01], among others). Qualified number restrictions are already supported by widely-used reasoners, including Racer and Fact [HM01, Hor98], and interface support has been incorporated into Protégé-OWL (although this extension technically produces non-conformant OWL, and is engineered with the expectation that QNRs will soon be added to the language).

There are a number of well-known partial workarounds which attempt to simulate aspects of qualified number restrictions in languages which do not support them (see, e.g., `http://www.cs.vu.nl/~guus/public/qcr.html`), however none are general solutions and determination of applicability can be problematic.

3.1.3 Use cases benefitting from the extension

The use cases presented in 2.1 and 2.9 both make heavy use of qualified number restrictions, and this deficiency of current languages has caused problems in a large number of otherwise straightforward semantic web modelling efforts. Examples include:

- Anatomy:
  “The normal hand has exactly five fingers of which one is a thumb.”
  “The heart has four chambers: two atria and two ventricles.”

- Bio-ontologies and chemistry:
  “Tricarboxylic acid contains exactly three carboxyl groups and one acidic group.”
  “Haemoglobin consists of four subunits, each of which contains exactly one haem group, each of which contains exactly one iron ion.”

- Many legal strictures, e.g. the British Nationality Act:
  An important concept in the act is of a "Person who has at least one parent who is a British citizen".

- Administrative structures:
  "A medical oversight committee must consist of at least five members of which two must be medically qualified, one a manager, and two members of the public."

- Drug interactions:
  "A legal drug regimen must not contain more than one Central Nervous System depressant, although it may contain any number of drugs in total."

- N-ary relations:
  "A person may have any number of 'features' or 'qualities', but (at a given time) only one of each: e.g. one 'height', one 'weight', one 'body temperature', etc. each of which can have a value, an uncertainty on that value, and a measure of its change or trend."
  "An employee may have many relationships with other employees, but only one line manager relationship."

3.2 Uncertainty handling

3.2.1 Definition of the requirement

One important requirement for ontology and rule languages, that follows from several real life applications (see for example the usecases in sections 2.2 and 2.3) is the ability to represent imprecise and vague knowledge and perform complex reasoning tasks with it. In such fuzzy extensions the syntax of the classical (crisp) DL language mainly remains unchanged, while classical assertions are extended to *fuzzy assertions* and interpretations are extended to *fuzzy interpretations*. More precisely, every assertion of the form $a : C$ now also carries the membership degree of the individual a to the *fuzzy concept* C, thus speaking of fuzzy assertions and fuzzy ABoxes. Returning to the brain anatomy example (2.2) one could define the following fuzzy knowledge base:

$$
\begin{aligned}
\mathcal{T} &= \{\mathsf{OPIFGyrus} \sqsubseteq \exists \mathsf{isDAPartOf}.\mathsf{IFGyrus}, \\
&\quad \mathsf{IFGyrus} \sqsubseteq \exists \mathsf{isDAPartOf}.\mathsf{FrontalLobe}, \\
&\quad \top \sqsubseteq \leq 1\, \mathsf{Inv}(\mathsf{isDAPartOf})\}, \\
\mathcal{R} &= \{\mathsf{Trans}(\mathsf{isAPartOf}), \mathsf{isDAPartOf} \sqsubseteq \mathsf{isAPartOf}\}. \\
\mathcal{A} &= \{(\langle \mathsf{o}_1, \mathsf{o}_2 \rangle : \mathsf{isDAPartOf}) \geq 0.8, \\
&\quad (\langle \mathsf{o}_2, \mathsf{o}_3 \rangle : \mathsf{isDAPartOf}) \geq 0.9, \\
&\quad (\langle \mathsf{o}_2', \mathsf{o}_3 \rangle : \mathsf{isDAPartOf}) \geq 0.3, \\
&\quad (\mathsf{o}_2 : \mathsf{IFGyrus}) \geq 0.85, (\mathsf{o}_1 : \mathsf{OPIFGyrus}) \geq 0.75, \\
&\quad (\mathsf{o}_3 : \leq 1\, \mathsf{Inv}(\mathsf{isDAPartOf})) \geq 0.7\}
\end{aligned}
$$

where $\mathcal{T}$ is a TBox, $\mathcal{R}$ is a RBox and $\mathcal{A}$ is a fuzzy ABox. As we can see, the syntactic changes indeed involve only fuzzy assertions and not the axioms of the TBox and RBox. The semantics of such definitions are based on fuzzy interpretations. More formally, a fuzzy interpretation comprises an *interpretation domain* Δ and an *interpretation function* $\cdot^{\mathcal{I}}$. The difference is that the interpretation function maps

1. an individual name $\mathsf{a} \in \mathbf{I}$ to an elements $\mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$,
2. a concept name $\mathsf{A} \in \mathbf{C}$ to a membership function $\mathsf{A}^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$,
3. a role name $R \in \mathbf{R}$ to a membership function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$.

For example, if $o \in \Delta^{\mathcal{I}}$ then $\mathsf{A}^{\mathcal{I}}(o)$ gives the degree to which the object o belongs to the fuzzy concept A, e.g. $\mathsf{A}^{\mathcal{I}}(o) = 0.8$. The interpretation function can be extended to give meaning to complex concepts and roles as well as to concept and role axioms [PFT$^+$04a, PFT$^+$05]. For example, a fuzzy interpretation $\mathcal{I}$ satisfies $\mathsf{A} \sqsubseteq C$ if $\forall o \in \Delta^{\mathcal{I}}, \mathsf{A}^{\mathcal{I}}(o) \leq C^{\mathcal{I}}(o)$ and it satisfies a fuzzy assertion of the form $a : C \geq 0.5$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq 0.5$.

The case of geospatial data, examined in section 2.3, is equivalent. In that case one can provide a fuzzy partition of all possible distances of locations, which will determine the membership degree of a location to the fuzzy concept “close” or “far”. For example, a fuzzy partition of distances might suggest that a location which distance is 97 meters is considered to be 0.9 “close”, thus writing $loc_1 : close \geq 0.9$.

Regarding the definition of rules, their syntax also remains unchanged. On the other hand the use of fuzzy concepts and roles in the atoms of a rule requires the extension of their semantics. Following [PSS$^+$], we say that a fuzzy interpretation $\mathcal{I}$ satisfies a rule of the form $\mathsf{Body}(\vec{\mathsf{x}}) \to \mathsf{Head}(\vec{\mathsf{x}})$, if $\mathsf{Body}^{\mathcal{I}}(\vec{\mathsf{x}}^{\mathcal{I}}) \leq \mathsf{Head}^{\mathcal{I}}(\vec{\mathsf{x}}^{\mathcal{I}})$.

Since the extension should not change the nature of the language and should coincide with the original language in cases that is not useful, it is important that it should generalize the two-valued Boolean logic of {0,1} into the interval [0,1], by providing a sound extension of Boolean logic. Hence, such a feature should not affect applications that do not require the specification of uncertainty.

3.2.2 Background work

Several extensions of ontology and rule languages have been proposed in the literature which extend the truth value set of {0,1} to [0,1] and covering different types of uncertainty. Some recent approaches on expressive fuzzy Description Logic languages have been presented in [Str05, SST$^+$05a, SST$^+$05b]. In [SST$^+$05a, SST$^+$05b] detailed reasoning algorithms for the fuzzy $\mathcal{SHOIN}$ and fuzzy OWL DL languages have been presented while a Fuzzy Reasoning Enginee (FiRE) is under implementation and optimization. Currently FiRE, `http://www.image.ece.ntua.gr/~nsimou/`, supports the fuzzy DL language f_{KD}-$\mathcal{SHI}$ [SST$^+$05a]. Fuzzy extensions on Semantic Web rule languages have also been studied in the literature. Some approaches can be found in [PSS$^+$], which considers the fuzzy SWRL language, and in [SSTP05] where a fuzzy extension of the RuleML framework is presented. Furthermore, a technical group on Fuzzy RuleML has recently been established (see `http://www.image.ntua.gr/FuzzyRuleML` for more details).

3.2.3 Use cases benefitting from the extension

The use cases described in sections 2.2 and 2.3 are both dependent on robust fuzzy reasoning support, and uncertainty and confidence extensions could benefit the semantic matching aspects of the "Recruitment from Worldwidejobs" use case [LN05, pages 6-11] noted in deliverable 1.1.4. Just as importantly, mature fuzzy reasoning extensions could allow the use of Description Logics in a wide range of novel applications currently employing less formal or heuristic reasoning systems.

3.3 User-defined Datatypes and Datatype Predicates

3.3.1 Definition of the requirement

The use case presented in Section 2.4 suggests that user-defined datatypes and datatype predicates are very useful in real-world ontology applications.

A *datatype* d is characterised by a lexical space, $L(d)$, which is an non-empty set of Unicode strings; a value space, $V(d)$, which is an non-empty set, and a total mapping $L2V(d)$ from the lexical space to the value space. For example, "boolean" is a datatype with value space $\{true, false\}$, lexical space {"true", "false","1","0"} and lexical-to-value mapping $\{$"true"$\mapsto true$,"false"$\mapsto false$,"1"$\mapsto true$,"0"$\mapsto false\}$. "$true$"^^xsd:boolean is a typed literal, while "true" is a plain literal.

A *datatype predicate* (or simply *predicate*) p is characterised by an arity $a(p)$, or a minimum arity $a_{min}(p)$ if p can have multiple arities, and a predicate extension (or simply *extension*) $E(p)$. The notion of predicate maps can be defined in an obvious way. For example, $=^{int}$ is a (binary) predicate with arity $a(=^{int}) = 2$ and extension $E(=^{int}) = \{\langle i_1, i_2 \rangle \in V(\mathtt{integer})^2 \mid i_1 = i_2\}$, where $V(\mathtt{integer})$ is the value space for the datatype `integer`.

In short, this requirement is asking for extending OWL DL with some new concept constructors so that user-defined datatypes and datatype predicates can be used in concept descriptions in the extended ontology language.

3.3.2 Background work

The concrete domain approach [BH91, Lut01] provides a rigorous treatment of named datatype predicates, rather than datatypes.[1] In the type system approach [HS01], datatypes are considered to be sufficiently structured by type systems; however, it does not specify how the derivation mechanism of a type system affects the set of datatypes **D**. [PH05b] proposes OWL-Eu, an extension of OWL DL that supports customised datatypes. The underpinning of OWL-Eu is the $\mathcal{SHOIN}(\mathcal{G}_1)$ DL, a combination of $\mathcal{SHOIN}$ and a unary datatype group. OWL-Eu is decidable if the combined unary datatype group is conforming; conformance of a unary datatype group precisely specifies the conditions on the set of supported datatypes. OWL-Eu provides a general framework for integrating OWL DL with customised datatypes, such as XML Schema non-list simple types. [PH05a] further further extends OWL-Eu to OWL-E, i.e. the $\mathcal{SHOIN}(\mathcal{G})$ DL (also decidable), to support user-defined datatype predicates.

3.3.3 Use cases benefitting from the extension

The datatype expressiveness provided in the official OWL specification is so limited that datatypes are often considered to lie entirely beyond OWL, to the extent that while the paucity of datatypes receives informal comment from nearly all practitioners it is seldom explicitly mentioned in use cases. The use case presented in 2.4 centers on user-defined datatypes, and one would expect a majority of the remaining use cases to also benefit from a richer datatype language: 2.3 would make use of geospatial datatypes, web service descriptions (2.5) could benefit from better descriptions of I/O datatypes, etc. The industrial use case "B2C portals from France Telecom" [LN05, pages 12-17] would also appear to benefit from geospatial and temporal datatype extensions.

[1]The reader is referred to Section 5.1.3 of [Pan04] for detailed discussions on concrete domains.

3.4 Closed World Assumption

3.4.1 Definition of the requirement

The unrestricted Open World Assumption employed by most current description formalisms is not always desirable. It sometimes useful to be able to specify a local Closed World Assumption and also to state negative queries for information with respect to closed sets of information (scoped negation as failure).

(Scoped) Negation As Failure While OWL allows the expression of axioms which cause instances to be classified as members of a class if some criteria can be proven to be fulfilled, it does not include an operate specifying that instances should be considered members of a class unless some criteria can be proven false. This is a special form of the "negation as failure" construct.

Integrity Constraints Language features have been proposed to express conditions describing when a particular closed knowledge base (KB) is not considered to be valid. Note that this notion of validity is different from traditional Description Logic satisfiability. Such integrity constraints could be provided via a special language construct (e.g. a logic programming rule with an empty head), that indicates when a constraint is violated.

Let consider a Semantic Web Service scenario in which a service allows as an input a search request with an author or keywords specified, but not both.

In a syntax borrowed from F-Logic this could be expressed by integrity constraints formalized as a logic programming rule with an empty head:

```
!- X:searchRequest[author->Y,keyword->Y].
```

This approach for expressing integrity constraints is adopted in two recent proposals for Semantic Web Service languages, namely WSML [dBFK+05] and SWSL [BBB+05]. Note that in many cases if you want to express integrity constraints in Logic Programming, this involves the use of negation as failure since the set of data to be checked is intended to be explicitly scoped. It has been proposed to formalize this such that the requirement "at least an author or a keyword needs to be present" by the following combination of LP rules and integrity constraints:

```
hasKey(X) :- X[author->Y].
hasAuthor(X) :- X[author->Y].
!- X:searchRequest, naf haskey(X), naf hasAuthor(X).
```

where `naf` stands for negation as failure.

3.4.2 Background work

In order to follow this requirement for expressing closed world reasoning using negation as failure and integrity constraints, the extension and integration of ontology languages with logic programming based rule languages have been proposed in the literature [dBFK+05, ABdB+05, BBB+05].

3.4.3 Use cases benefitting from the extension

The use cases presented in 2.5 and 2.7 both offer examples of OWL's open world assumption causing usability problems, and it can be argued that the non-specialist usage scenarios from 2.8 highlight the confusion which can occur among users unfamiliar with OWL's formal semantics and more comfortable working with closed-world systems. Whether provided through tightly integrated language extensions, query languages, or workarounds which sit entirely above OWL's semantic model, a simple way to limit application of the open world assumption could be of benefit in a number of scenarios.

3.5 Integration with Logic Programming

3.5.1 Definition of the requirement

Following on from the above requirement, value can be seen in knowledgebases expressed using combinations of different formalisms (i.e. combining first order logic with aspects of logic programming).

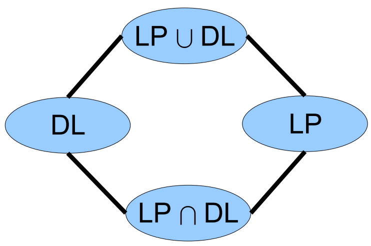

Figure 3.1: Integration of LP and FOL

Figure 3.1 shows the layering having as lowest layer the semantic intersection of Logic Programming (LP) and Description Logic Semantics (DL). The upper oval indicates the semantic union of LP and DL. Here several open questions have to be resolved. We give at the moment only some concrete examples:

Example 1 *Given an FOL theory* Φ*:* $\exists x.p(x)$

And given a Logic Program P*:* $q(x) \leftarrow p(x)$

What are the entailments of the combined theory $\Phi \uplus P$? More specifically, can we conclude $\exists x.q(x)$? If we were to interpret the logic program as a first-order theory and we were to interpret $\leftarrow$ as material implication, one can say that $\exists x.q(x)$ can be concluded. If we take the LP point-of-view, however, we can say that we can only effectively talk about known individuals and thus this conclusion is not warranted.

Example 2 *Given an FOL theory Φ: $a = b$*

And given a Logic Program P: $p(a)$ $p(c) \leftarrow p(a), p(b)$

What are the entailments of the combined theory $\Phi \uplus P$? More specifically, can we conclude $p(c)$? Or rather, does this lead to an inconsistency because of the unique names assumption in P? Intuitively, we would expect $p(c)$ to be entailed, since $a = b$ is in the FOL theory.

Example 3 *Given an FOL theory Φ: $r \supset p$*

And given a Logic Program P: r
$q \leftarrow p$

What are the entailments of the combined theory $\Phi \uplus P$? More specifically, can we conclude q? Intuitively, we would expect q to be entailed, since r is in P and $r \supset p$ is in Φ.

Example 4 *Given an FOL theory Φ: $q \supset r$*

And given a Logic Program P: $q \leftarrow not\ p$

What are the entailments of the combined theory $\Phi \uplus P$? More specifically, can we conclude r? Intuitively, we might expect r to be entailed, since q is a conclusion of P and if q is true, the r is true in Φ. However, the conclusion q is based on a weak form of negation, and thus one might argue that q cannot be used in Φ to conclude additional facts.

Example 5 *Given an FOL theory Φ: $\forall x.p(x)$*

And given a Logic Program P: $q(x) \leftarrow p(x)$
$r(x) \leftarrow$ *not* $q(x)$

What are the entailments of the combined theory $\Phi \uplus P$? More specifically, can we conclude $\forall x.q(x)$? Alternatively, can we conclude $\neg\exists x.r(x)$ Intuitively, we might expect $\forall x.q(x)$ to be entailed, since $p(x)$ holds for all x and there is an LP which which says "if $p(x)$ then $q(x)$". To find out whether $\neg\exists x.r(x)$ it is important to know what the interaction is between the classical negation in the FOL theory and the negation-as-failure in the Logic Program.

Basically, the requirement for a unified semantic framework for FOL- and LP-based languages should at least provide "intuitively correct" semantics to the above mentioned examples. A more concrete definition of requirements is the subject of ongoing work, e.g. [dB05].

3.5.2 Background work

Initial work on achieving such a unified semantic framework has been conducted in the WSML working group [dB05].

3.5.3 Use cases benefitting from the extension

A logic programming extension could be used to provide negation-as-failure, so use cases 2.5 and 2.7 could benefit. The "Product Lifecycle Management" industrial use case [LN05, 24-27] explicitly cites such extensions as potentially beneficial. Integration with relatively popular logic programming systems could also expose a wider community with a huge range of "traditional" programming chores to semantic web languages, and allow a much easier transition from classic procedural implementation to semantic description-based applications.

3.6 Non Semantic Requirements

3.6.1 Definition of the requirement

There are a number of features which which we do not find in current language proposals for the Semantic Web.

One syntactic framework for a set of layered languages We believe different Semantic Web and Semantic Web Service applications need languages of different expressiveness. There already exist language recommendations for certain aspects, such as the Ontology languages RDFS [Bese04] and OWL [DS04]. Already in the case of RDFS and OWL, we can see that layering languages on existing recommendations is not straightforward [HPSvH03, dBPLF05]; either the layering is not strict, or certain desirable features of a language, such as the ability to use existing efficient reasoners, are lost.

Normative, human readable syntax It has been argued that tools will hide language syntax from the user; however, as has been seen with the adoption of SQL, an expressive but understandable syntax is crucial for successful adoption of a language. Developers and early adopters of the language will have to deal with the concrete syntax. If it is easy to read and understand it will allow for easier adoption of the language.

Separation of conceptual and logical modeling On the one hand, the conceptual syntax of WSML has been designed in such a way that it is independent of the underlying logical language and no or only limited knowledge of formal languages is required for the basic modeling of Web Services, Goals, Mediators and Ontologies. On the other hand, the logical expression syntax allows expert users to refine definitions on the conceptual syntax using the full expressive power of the underlying logic, which depends on the particular language variant chosen by the user.

Semantics based on well known formalism Well known logical formalisms such as Datalog and Description Logics should be reused while maintaining the established computational properties of the original formalisms. Most importantly, this allows the reuse of tools already developed for these formalisms.

Frame-based syntax Frame Logic [KLW95] allows the use of frames in logical expressions. This allows the user to work directly on the level of concepts, attributes, instances, and attribute values, instead of at the level of predicates. Furthermore, variables are allowed in place of concept and attribute identifiers, which enables meta-modeling and reasoning over the signature.

3.6.2 Background work

Basically all work on Description Logics, Logic Programming and integration of both is relevant.

RDFS RDFS [Bese04] is a simple ontology modeling language based on triples. It allows to express classes, properties, class hierarchies, property hierarchies, and domain- and range restrictions. Several proposals for more expressive Semantic Web and Semantic Web Service descriptions extend RDFS, however there are difficulties in semantically layering an ontology language on top of RDFS:

1. RDFS allows the use of the language vocabulary as subjects and objects in the language itself.

2. RDFS allows the use of the same identifier to occur at the same time in place of a class, individual, and property identifier.

We believe that the number of use cases for the first feature, namely the use of language constructs in the language itself, is limited. However, the use of the same identifier as class, individual and property identifier (also called meta-modeling) is useful in many cases [Sch02, dBPLF05]. WSML does not allow the use of the language constructs in arbitrary places in an ontology, but does allow meta-modeling in its Flight, Rule and Full variants.

OWL The Web Ontology Language OWL [DS04] is a language for modeling ontologies based on the Description Logic paradigm. OWL consists of three species, namely OWL Lite, OWL DL and OWL Full, which are intended to be layered according to increasing expressiveness. OWL Lite is a notational variant of the Description Logic $\mathcal{SHIF}(\mathbf{D})$; OWL DL is a notational variant of the Description logic $\mathcal{SHOIN}(\mathbf{D})$ [HPSvH03]. It turns out that OWL DL adds very little in expressiveness to OWL Lite [HPSvH03]. The most expressive species of OWL, OWL Full, layers on top of both RDFS and OWL DL, and because these languages are so different, the semantics of OWL Full is not straightforward and is not a proper extension of the OWL DL semantics [dBPLF05].

3.6.3 Use cases benefitting from the extension

Usability concerns underlie all usage scenarios. The primary motivation for improvements to ontology languages at this level is the realization that tools will never completely insulate implementors from the details of the language used, and users will be forced to deal with the more "mundane" aspects of ontology languages at some point.

3.7 Expressability of State relations in Bi-state rules for describing dynamic services

3.7.1 Definition of the requirement

Ontology languages could be extended with constructs to allow expression of dynamic changes to knowledge by the execution of services.

Such an extension should offer a clear syntax and semantics for defining pre-conditions and post-conditions, and it must be possible to express the relation between pre- and post-conditions guaranteed by a service's execution.

3.7.2 Background work

Preliminary work towards a syntax able to express state relations and effects of Web services has been conducted in the context of SWSL/SWSO[BBB$^+$05], WSMO[dBBD$^+$05, LK04], and OWL-S surface syntax [MBH$^+$04][2].

[2]The surface syntax is not yet part of the OWL-S W3C submission, but available as draft atwww.daml.org/services/owl-s/1.1/owl-s-gram.pdf

3.7.3 Use cases benefitting from the extension

Current work on such extensions is tightly focused on solving the specific problem of web service description, mentioned in 2.5. Web service descriptions are considered to be a major future use of semantic web languages, and issues in this domain are the primary focus of work package 2.4.

3.8 Graphical extensions of Ontology Tools

3.8.1 Requirements and opportunities

For an ontology language to be easily understood by domain experts:

- Its constructors should be close to the language that business experts speak and understand. For example, it is easier for domain experts to say that *"it is mandatory for each book to have an ISBN"*, than to say *"the cardinality between the concept book and the concept ISBN is (1:0)"*; or to say that a person is identified by his passport number than to say the cardinality between them is (1:1). Notice that the notions of "mandatory" and "identity" can be specified through the use of cardinality.
- It should have a graphical notation to enable quick and simple modelling. With a graphical notation here we do not mean merely a visualization, but a "graphical language" that allows domain experts to construct an ontology using a graphical notation for concept, relation, and axiom. In other words, such a language should guide experts to "think" conceptually while building an ontology. Notice that the users of current ontology modelling tools think outside these tools (maybe drawing on papers, etc.), and enter their results to the tools. Ontology visualization in such tools is very simplistic (e.g. only connection between concepts), and only used to view, rather than to model, knowledge.

Instead of inviting new ontology languages to meet such requirements, i.e. that business people can understand, it is worth investigating how existing business languages and graphical notations can be reused for ontology engineering purposes. Such languages could be the Object Role Modelling (ORM), UML diagrams (not only the class-diagram), etc[3].

Experiments on defining a full mapping of ORM into description logics is very promising[4]. See an example of an axiomatization built using ORM in figure 3.2. We have found

[3] Many other business notations could be found in Microsoft Visio.

[4] This experiment was done during a Kweb research exchange between the Free University of Brussels (VUB) and the Free University of Bolzano (FUB).

that almost[5] all ORM constructs can be mapped into description logics, and so to OWL-DL. We believe that such mapping will enable the inheritance and reuse of the ORM literature, tools, techniques, applications and users, etc.

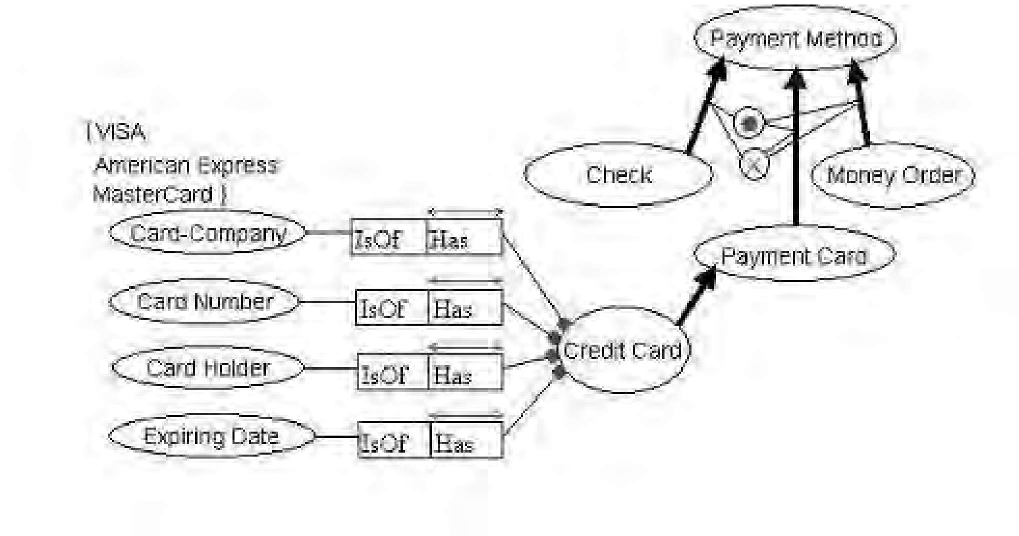

Figure 3.2: An example of ORM model.

ORM is being used by business experts, especially in the database and business rules communities. ORM is a successor of the NIAM (Natural-language Information Analysis Method, developed in the early '70s). It was explicitly designed to play the role of a stepwise methodology: to arrive at the "semantics" of a business application's data based on natural language communication between domain experts. See for example the verbalization of the "subset" constraint in figure X2, which is generated automatically. ORM is not only a conceptual modelling language but also comprehensive in its treatment of many "practical" or "standard" business rules and constraint types (e.g. identity, mandatory, uniqueness, subsumption, subset, equality, exclusion, value, frequency, symmetric, intransitive, acyclic, etc.). Furthermore, ORM has an expressive and stable graphical notation since it captures many rules graphically and it minimizes the impact of change on the models.

To sum up, the mapping of business languages and graphical notations into description logic based languages has the following immediate advantages:

1. It allows business domain experts to participate in the ontology modelling and validation processes;

[5]Except a few rarely used ORM constraint types such as external frequency, external uniqueness, and some of the ring constraints.

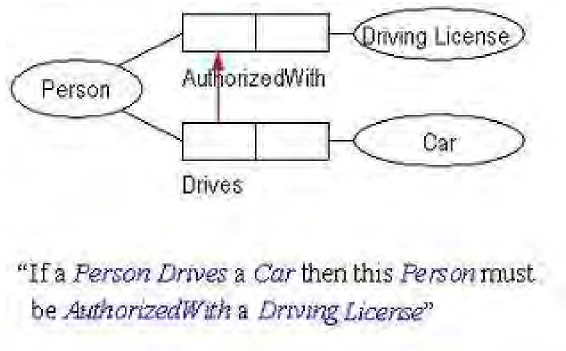

Figure 3.3: An example of a subset constraint verbalized (automatically) into a natural language sentence.

2. It enables the reuse of the business literature (such as tools, methods, applications, users, best practice, etc.) In other words, current ontology languages will be moved from the mathematical sense into "business logics";

3. It is a good way to test and improve description logic based languages themselves, as this mapping will discover which logical operators are needed and used in industry.

We shall continue to instigate and incorporate this kind of language extensions in the third and the forth JPAs of WP2.5.

3.8.2 Use cases benefitting from the extension

Knowledge solicitation from business and domain experts is a crucial aspect of all significant ontology projects, and it is clear that multiple levels of fluency in ontology languages will persist. The use cases presented in 2.8 and 2.10 focus specifically on the various actors involved in the creation and use of ontologies, but in practice the vast majority of ontology construction is likely to benefit from tools which make knowledge modelling easier for users.

3.9 Modularizing and Aligning Ontologies

3.9.1 Definition of the requirement

As the use case in Section 2.10 demonstrates, the domain sometimes need to be modularized into different viewpoints. Each expert prefers to use her own viewpoint with her own terminologies. Modelling viewpoints can be done with current ontology languages like OWL.

However, all standardized ontology languages lack a means of aligning these local ontologies in an appropriate way. Integrating all local ontologies into on global ontology may have some unwanted effects on the global ontology, including unnatural and unusual subsumption hierarchies and/or inconsistencies. So we need an extension of ontology languages which allows the representation of bridges between concepts of different (local) ontologies.

3.9.2 Background work: Contextualizing ontologies

A lot of work has been done in the past investigating modularization (see for example KWEB deliverable D2.1.3.1: Report on modularization of ontologies) and Ontology Matching (see for example D2.2.4: Description of alignment implementation and benchmarking results). For aligning complex ontologies a possible solution may be an extension known as C-OWL.

C-OWL

C-OWL is an extension of OWL for representing *contextualized (or contextual) ontologies* [BGvH$^+$04]. Contextualized ontologies are local representations of a domain—named contexts—that are semantically related with other contexts thanks to mappings. The original motivation for C-OWL was the alignment and coordinated use of ontologies made for different purposes. In our framework, C-OWL is used as a way of formalizing and implementing several alternative representations of the domain that we call *viewpoints*. In C-OWL, the knowledge about a domain is contained in a set of contexts. Each context O_i is an OWL ontology, with its own language and its own interpretation. Mappings are expressed by bridge rules. A bridge rule from O_i to O_j is a way to declare a correspondence between the interpretation domains of these two contexts. On the basis of these correspondences, a part of the knowledge contained in O_i can be interpreted and reused in O_j.

Formally, a C-OWL *context space* contains a set of contexts $\{O_i\}_{i \in I}$, I being a set of indexes for contexts. The indexes of I are used to prefix the expressions, associating an expression with the context in which it is defined. For example, `i:C`, `i:`$\exists$`p.C`, `i:a`, `i:C` $\sqsubseteq$ `D` and `i:C(a)` are expressions of the local language of O_i.

The semantics of a context space are given by a distributed interpretation $\mathfrak{I}$ that contains an interpretation $\mathcal{I}_i$ for each $i \in I$. Each $\mathcal{I}_i$ is composed of a local interpretation domain $\Delta^{\mathcal{I}_i}$ and a local interpretation function $\cdot^{\mathcal{I}_i}$. A context is interpreted with the corresponding local interpretation, i.e. an axiom or an assertion of O_i is satisfied by $\mathfrak{I}$ if it is satisfied by $\mathcal{I}_i$.

A mapping $\mathcal{M}_{ij}$ is a set of *bridge rules* from O_i to O_j. There are different types of bridge rules, occurring between classes, individuals or properties of two contexts. An *into rule* is a bridge rule of the form $\mathtt{i:C} \xrightarrow{\sqsubseteq} \mathtt{j:D}$, where $\mathtt{i:C}$ and $\mathtt{j:D}$ are classes respectively from O_i and O_j. This rule means that the class $\mathtt{i:C}$ of O_i is considered, from the viewpoint of O_j, as more specific than the class $\mathtt{j:D}$ [ST04]. The *onto rule* $\mathtt{i:C} \xrightarrow{\sqsupseteq} \mathtt{j:D}$ means that O_j considers the class $\mathtt{i:C}$ to be more general than $\mathtt{j:D}$. Bridge rules are directional: a bridge rule from O_i to O_j is considered in the viewpoint of O_j, and thus, $\mathtt{i:C} \xrightarrow{\sqsubseteq} \mathtt{j:D}$ is not equivalent to $\mathtt{j:D} \xrightarrow{\sqsupseteq} \mathtt{i:C}$, where the latter says that the class $\mathtt{j:D}$ of O_j is considered more general, from the viewpoint of O_i, from the class $\mathtt{i:D}$ of O_i.

Formally, the distributed interpretation $\mathfrak{I}$ of a context space is associated with a set of *domain relations*. A domain relation $r_{ij} \subseteq \Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ states, for each object of $\Delta^{\mathcal{I}_i}$, the object of $\Delta^{\mathcal{I}_j}$ to which it corresponds. The notation $r_{ij}(\mathtt{C}^{\mathcal{I}_i})$ denotes the interpretation of the class $\mathtt{i:C}$ of O_i as considered in the interpretation domain of O_j. Then, the semantics of onto and into bridge rules are given with respect to domain relations: $\mathfrak{I}$ satisfies $\mathtt{i:C} \xrightarrow{\sqsubseteq} \mathtt{j:D}$ if $r_{ij}(\mathtt{C}^{\mathcal{I}_i}) \subseteq \mathtt{D}^{\mathcal{I}_j}$ and $\mathfrak{I}$ satisfies $\mathtt{i:C} \xrightarrow{\sqsupseteq} \mathtt{j:D}$ if $r_{ij}(\mathtt{C}^{\mathcal{I}_i}) \supseteq \mathtt{D}^{\mathcal{I}_j}$.

Another form of bridge rule is used to specify a correspondence between individuals. $\mathtt{i:a} \xrightarrow{\equiv} \mathtt{j:b}$ means that the individual $\mathtt{i:a}$ in O_i corresponds to the individual $\mathtt{j:b}$ in O_j. Formally, $\mathfrak{I}$ satisfies $\mathtt{i:a} \xrightarrow{\equiv} \mathtt{j:b}$ if $r_{ij}(\mathtt{a}^{\mathcal{I}_i}) = \mathtt{b}^{\mathcal{I}_j}$.

Global and Local Reasoning

Local reasoning services in C-OWL are the standard OWL reasoning services, performed in a particular context, without taking into account the bridge rules. A *global reasoning service* uses bridge rules to infer statements in a context using knowledge from the other contexts. [ST04] presents an extension of the standard tableau algorithm for the computation of the global subsumption test. *Global subsumption* uses the principle of subsumption propagation which, in its simplest form, can be expressed as:

if the mapping $\mathcal{M}_{ij}$ contains $\mathtt{i:A} \xrightarrow{\sqsupseteq} \mathtt{j:C}$ and $\mathtt{i:B} \xrightarrow{\sqsubseteq} \mathtt{j:D}$
then $\mathfrak{I}$ satisfies $\mathtt{i:A} \sqsubseteq \mathtt{B}$ implies that $\mathfrak{I}$ satisfies $\mathtt{j:C} \sqsubseteq \mathtt{D}$.

Intuitively, this means that subsumption in a particular context can be inferred from subsumption in another context thanks to bridge rules.

Similarly, we consider here *global instance checking* based on an instantiation propagation rule:

if $\mathcal{M}_{ij}$ contains $\mathtt{i:C} \xrightarrow{\sqsubseteq} \mathtt{j:D}$ and $\mathtt{i:a} \xrightarrow{\equiv} \mathtt{j:b}$
then $\mathfrak{I}$ satisfies $\mathtt{i:C(a)}$ implies that $\mathfrak{I}$ satisfies $\mathtt{j:D(b)}$.

Instantiation is extended in order to use global instance checking. Based on bridge rules, information known about an individual in a particular context can be completed using inferences made in other contexts.

3.9.3 Use cases benefitting from the extension

The use case presented in 2.10 makes use of extensive C-OWL functionality, however even simpler cases of information integration, such as that described in 2.7, could be implemented using the C-OWL extensions.

The article [BGvH+04] mentions an application of C-OWL, previously presented in [SvHSG04], and concerning the mapping of three ontologies in the medical domain, namely Galen [RN93], Tambis [BGB+99] and the semantic network of the UMLS [NP02]. Moreover, the present research work in the context of the KASIMIR project seems to be, at the moment, the unique application of C-OWL for the manipulation of viewpoints within a problem solving process.

Chapter 4

Conclusion

In this report, commonly-encountered limitations of the semantic web languages RDF and OWL have been investigated through study of ontology applications as they are being created and used in practice. A wide variety of issues have been identified, and it has been shown that many of these problems can be overcome by providing extensions to ontology languages. Proposed extensions run the range from theoretically minor modifications (syntactic changes and graphical representations), through well-studied enhancements to expressiveness (extended datatypes and qualified number restrictions), to major additions which could significantly impact the languages' underlying formal model in complex and subtle ways (negation-as-failure and logic programming extensions). The data presented here provide valuable data for evaluating the impact of extensions on real-world applications.

Some of the most common limitations have already been addressed in implemented systems with language extensions where these extensions are well understood, and efforts have already begun with a view to standardization of this "low-hanging fruit" in version 1.1 of OWL. More ambitious enhancements to OWL, and "extension" in the form of higher-level semantic languages, interfaces, and usage patterns continue to proceed with collaboration between researchers, implementers, and users.

Bibliography

[ABdB+05] Jürgen Angele, Harold Boley, Jos de Bruijn, Dieter Fensel, Pascal Hitzler, Michael Kifer, Reto Krummenacher, Holger Lausen, Axel Polleres, and Rudi Studer. Web rule language (WRL). W3C Member Submission, available from `http://www.w3.org/Submission/2005/08/`, June 2005.

[Act] Actkey. Web Page `http://flora.huh.harvard.edu:8080/actkey`.

[ASB+04] A. Alonso, J. Sasin, N. Bottini, I. Friedberg, I. Friedberg, A. Osterman, A. Godzik, T. Hunter, J. Dixon, and T. Mustelin. Protein tyrosine phosphatases in the human genome. *Cell*, 117(6):699–711, Jun 11 2004.

[BBB+05] Steve Battle, Abraham Bernstein, Harold Boley, Benjamin Grosof, Michael Gruninger, Richard Hull, Michael Kifer, David Martin, Deborah L. McGuinness, Sheila McIlraith, Glen Newton, David De Roure, Mark Skall, Jianwen Su, Said Tabet, and Hideki Yoshida. Semantic web services framework (SWSF). W3C Member Submission, available from http://www.w3.org/Submission/2005/07/, May 2005.

[BCF+03] Scott Boag, Don Chamberlin, Mary F. Fernández, Daniela Florescu, Jonathan Robie, and Jérôme Siméon (editors). XQuery 1.0: An XML Query Language. URL, November 12 2003. `http://www.w3.org/TR/2003/WD-xquery-20031112/`, W3C Working Draft.

[Bec04] Dave Beckett. RDF/XML Syntax Specification. URL, February 10 2004. W3C Recommendation, `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`.

[Ber94] Hollunder Bernhard. An alternative proof method for possibilistic logic and its application to terminological logics. In *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 327–335, San Francisco, CA, 1994. Morgan Kaufmann Publishers.

[Bese04] Dan Brickley, Ramanathan V. Guha (editor), and Brian McBride (series editor). RDF Vocabulary Description Language

1.0. URL, February 10 2004. W3C Recommendation, `http://www.w3.org/TR/2004/REC-rdf-schema-20040210/`.

[BGB+99] P.G. Baker, C.A. Goble, S. Bechhofer, N.W. Paton, R. Stevens, and A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6):510–520, 1999.

[BGvH+03] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. C-OWL: Contextualizing ontologies. In *The SemanticWeb - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*. Springer, 2003.

[BGvH+04] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing Ontologies. *Journal of Web Semantics*, 1(4):1–19, 2004.

[BH91] Franz Baader and Philipp Hanschke. A Schema for Integrating Concrete Domains into Concept Languages. pages 452–457, 1991.

[BPSM+04] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau (editors). Extensible Markup Language (XML) 1.0 (Third Edition). URL, February 04 2004. `http://www.w3.org/TR/2004/REC-xml-20040204`, W3C Recommendation.

[BS02] A. Borgida and L. Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In *Proc. of the Intenational Conference on Cooperative Information Systems*, 2002.

[BvHH+04] Sean Bechhofer, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein eds. OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/, Feb 2004.

[CFB05] H. Chen, S. Fellah, and Y. Bishr. Rules for geospatial semantic web applications. W3C Workshop on Rule Languages for Interoperability, 2005.

[Coh97] P. T. Cohen. Novel protein serine/threonine phosphatases: variety is the spice of life. *Trends Biochem Sci*, 22(7):245–51, July 1997.

[CTM87] A.R. Clapham, T.G. Tutin, and D.M. Moore. *Flora of the British Isles*. Cambridge University Press, 1987.

[dB05] Jos de Bruijn. Ontology languages around fol and lp. Deliverable D28.3, WSML Working Group, October 2005. Available from `http://www.wsmo.org/TR/d28/d28.3/`.

[dBBD+05] Jos de Bruijn, Christoph Bussler, John Domingue, Dieter Fensel, Martin Hepp, Uwe Keller, Michael Kifer, Birgitta König-Ries, Jacek Kopecky, Rubén Lara, Holger Lausen, Eyal Oren, Axel Polleres, Dumitru Roman, James Scicluna, and Michael Stollberg. Web service modeling ontology (WSMO). W3C Member Submission, available from http://www.w3.org/Submission/2005/06/, April 2005.

[dBFK+05] Jos de Bruijn, Dieter Fensel, Uwe Keller, Michael Kifer, Holger Lausen, Reto Krummenacher, Axel Polleres, and Livia Predoiu. Web service modeling language (WSML). W3C Member Submission, available from http://www.w3.org/Submission/2005/06/, April 2005.

[dBLN04] M. d'Aquin, S. Brachais, J. Lieber, and A. Napoli. Decision Support and Knowledge Management in Oncology using Hierarchical Classification. In *Proc. of the Symposium on Computerized Guidelines and Protocols, CGP-2004*, volume 101 of *Studies in Health Technology and Informatics*, pages 16–30. IOS Press, 2004.

[dBPLF05] Jos de Bruijn, Axel Polleres, Rubén Lara, and Dieter Fensel. OWL DL vs. OWL Flight: Conceptual modeling and reasoning for the semantic web. In *Proceedings of the 14th International World Wide Web Conference (WWW2005)*, 2005.

[DGM04] Olivier Dameron, Bernard Gibaud, and Mark Musen. Using semantic dependencies for consistency management of an ontology of brain-cortex anatomy. In *First International Workshop on Formal Biomedical Knowledge Representation KRMED04*, pages 30–38, 2004.

[DP04] Zhongli Ding and Yun Peng. A Probabilistic Extension to Ontology Language OWL. In *Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37).*, page 10, Big Island, Hawaii, January 2004.

[DS04] Mike Dean and Guus Schreiber, editors. *OWL Web Ontology Language Reference*. 2004. W3C Recommendation 10 February 2004.

[FHe03] Richard Fikes, Patrick Hayes, and Ian Horrocks (editors). DAML Query Language (DQL) Abstract Specification; Joint United States/European Union ad hoc Agent Markup Language Committee; DARPA Agent Markup Language (DAML) Program, April 1 2003. `http://www.daml.org/2003/04/dql/`.

[FHH03] Richard Fikes, Patrick Hayes, and Ian Horrocks. OWL-QL - A Language for Deductive Query Answering on the Semantic Web. Technical report, Knowledge Systems Laboratory, Stanford University, Stanford, CA, KSL-03-14, 2003.

[GBDG05] C. Golbreich, O. Bierlaire, O. Dameron, and B. Gibaud. Use case: Ontology with rules for identifying brain anatomical structures. W3C Workshop on Rule Languages for Interoperability, 2005.

[GCWH+99] Hillman G.R., Chang C-W., Ying H., Yen J., Ketonen L., and Kent T.A. A fuzzy logic approach to identifying brain structures in mri using expert anatomic knowledge. *Computers and Biomedical Research*, 32:503–516, 1999.

[Gie03] J. Gielis. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany*, 90:333–338, 2003.

[GL02] Rosalba Giugno and Thomas Lukasiewicz. P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, pages 86–97, London, UK, 2002. Springer-Verlag.

[Gle63] H.A. Gleason. *The New Britton and Brown Illustrated Flora of the North-eastern United States and Adjacent Canada*. Hafner Publishing Company, New York, 1963.

[Gol05] Christine Golbreich. What reasoning support for ontology and rules? the brain anatomy case study. In *Proceedings of the 2005 OWL Workshop*, Nov 2005.

[Gua97] N. Guarino. Understanding, Building, and Using Ontologies. *International Journal of Human-Computer Studies (IJHCS)*, 46(2–3):293–310, 1997.

[HM01] Volker Haarslev and Ralf Möller. RACER system description. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-01)*, volume 2083 of *Lecture Notes in Artificial Intelligence*, pages 701–705. Springer-Verlag, 2001.

[Hor98] I. Horrocks. Using an Expressive Description Logic: FaCT or Fiction? In *International Conference of Knowledge Representation (KR98)*, pages 636–647, 1998.

[HPSB+04] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A Semantic Web Rule Language — Combining OWL and RuleML. W3C Member Submission, `http://www.w3.org/Submission/SWRL/`, May 2004.

[HPSvH03] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[HS01] Ian Horrocks and Ulrike Sattler. Ontology reasoning in the $\mathcal{SHOQ}$(D) description logic. In *Proc. of International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 199–204, 2001.

[HSLS+04] N. Hulo, C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. Recent improvements to the prosite database. *Nucleic Acids Res*, 32(Database issue):D134–7, Jan 1 2004.

[KAC+02] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the eleventh international conference on World Wide Web*, pages 592–603, Honolulu, Hawaii, USA, May 7–11 2002. ACM Press, New York, USA. ISBN:1-58113-449-5.

[Ken01] P. J. Kennelly. Protein phosphatases–a phylogenetic perspective. *Chem Rev*, 101(8):2291–312, August 2001.

[KLW95] Michael Kifer, Geord Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *JACM*, 42(4):741–843, 1995.

[KY95] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall, 1995.

[LCS+04] I. Letunic, R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork. Smart 4.0: towards genomic data integration. *Nucleic Acids Res*, 32(Database issue):D142–4, Jan 1 2004.

[LGMR01] Paul A. Longley, Michael F. Goodchild, David J. Maguire, and David W. Rhind. *Geographic Information Systems and Science*. John Wiley and Sons, 2001.

[LK04] Holger Lausen and Uwe Keller. Functional description of web services. Deliverable D28.1, WSML Working Group, 2004. Available from `http://www.wsmo.org/TR/d28/d28.1/`.

[LN05] Alain Léger and Lyndon Nixon. System and knowledge technology components for prototypical applications and business cases. Kweb deliverable: D1.1.4, 2005.

[LR99] Ora Lassila and Ralph R.Swick. Resource Description Framework (RDF) Model and Syntax Specification – W3C Recommendation 22 February 1999. Technical report, World Wide Web Consortium, 1999.

[Lut01] Carsten Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, RWTH Aachen, 2001.

[MAA+05] N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, P. Bork, P. Bucher, L. Cerutti, R. Copley, E. Courcelle, U. Das, R. Durbin, W. Fleischmann, J. Gough, D. Haft, N. Harte, N. Hulo, D. Kahn, A. Kanapin, M. Krestyaninova, D. Lonsdale, R. Lopez, I. Letunic, M. Madera, J. Maslen, J. McDowall, A. Mitchell, A. N. Nikolskaya, S. Orchard, M. Pagni, C. P. Ponting, E. Quevillon, J. Selengut, C. J. Sigrist, V. Silventoinen, D. J. Studholme, R. Vaughan, and C. H. Wu. Interpro, progress and status in 2005. *Nucleic Acids Res*, 33(Database issue):D201–5, Jan 1 2005.

[MBH+04] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic markup for web services. W3C Member Submission, available from http://www.w3.org/Submission/2004/07/, November 2004.

[NP02] Stuart J. Nelson and Betsy L. Powell, Tammy andn Humphreys. The unified medical language system (umls) project. In Allen Kent and Carolyn M. Hall, editors, *Encyclopedia of Library and Information Science.*, pages 369–378. Marcel Dekker, Inc., 2002.

[Pan04] Jeff Z. Pan. *Description Logics: Reasoning Support for the Semantic Web*. PhD thesis, School of Computer Science, The University of Manchester, 2004.

[Pe04] Eric Prud'hommeaux and Andy Seaborne (editors). SPARQL Query Language for RDF (Working Draft). URL, October 2004. `http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/`, W3C Working Draft.

[PFT+04a] J. Pan, E. Franconi, S. Tessaris, Stamou, V. Tzouvaras, L. Serafini, I. Horrocks, and B. Glimm. Specification of coordination of rule and ontology. Kweb deliverable: D2.5.1, 2004.

[PFT+04b] Jeff Z Pan, Enrico Franconi, Sergio Tessaris, Birte Glimm, Wolf Siberski, Vassilis Tzouvaras, Giorgos Stamou, Ian Horrocks, Lei Li, and Holger Wache. D252 report on query language design and standardisation. KnowledgeWeb Deliverable KWEB/2004/D252/v10, University of Manchester, 2004.

[PFT+05] J. Pan, E. Franconi, S. Tessaris, D. Tsarkov, I. Horrocks, G. Stoilos, G. Stamou, H. Wache, D. Turi, S. Bechhofer, and L. Li. Report on inplementataion and optimisation techniques for ontology query systems. Kweb deliverable: D2.5.3, 2005.

[PH03a] Jeff Z. Pan and Ian Horrocks. RDFS(FA): A DL-ised Sub-language of RDFS. In *Proc. of the 2003 International Workshop on Description Logics (DL2003)*, 2003.

[PH03b] Jeff Z. Pan and Ian Horrocks. Web Ontology Reasoning with Datatype Groups. In *Proc. of the 2nd International Semantic Web Conference (ISWC2003)*, 2003.

[PH05a] Jeff Z. Pan and Ian Horrocks. Introducing Customised Datatypes and Datatype Predicates into OWL. In *Proc. of the International workshop on OWL: Experience and Directions (OWL-ED2005)*, 2005. To appear.

[PH05b] Jeff Z. Pan and Ian Horrocks. OWL-Eu: Adding Customised Datatypes into OWL. In *Journal of Web Semantics*, 2005. To appear.

[PSS$^+$] J.Z. Pan, G. Stoilos, G. Stamou, V. Tzouvaras, and I. Horrocks. f-swrl:a fuzzy extension of swrl. *Data Semantics, Accepted*.

[RBG] Kew Royal Botanic Gardens. electronic plant information centre. Published on the Internet `http://www.kew.org/epic/`.

[RN93] A.L. Rector and W.A. Nowlan. The galen project. *Computer Methods and Programs in Biomedicine*, 45:75–78, 1993.

[(RS05a] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie chimiothérapie néo-adjuvante. `http://www.oncolor.org/referentiels/sein/sein_ttt_adj_neo.htm`, 2005. dernière consultation : mai 2005.

[(RS05b] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie chirurgie des tumeurs opérables d'emblée. `http://www.oncolor.org/referentiels/sein/sein_ttt_chir_arb.htm`, 2005. dernière consultation : mai 2005.

[(RS05c] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie radiothérapie. `http://www.oncolor.org/referentiels/sein/sein_ttt_rdth_arb.htm`, 2005. dernière consultation : mai 2005.

[(RS05d] Réseau de santé Oncolor (RSO). Référentiel pour le diagnostic et le traitement du cancer du sein – partie traitement médical adjuvant. `http://www.oncolor.org/referentiels/sein/sein_ttt_adj.htm`, 2005. dernière consultation : mai 2005.

[Sch02] Guus Schreiber. The web is not well-formed. *IEEE Intelligent Systems*, 17(2), 2002. Contribution to the section Trends and Controversies: Ontologies KISSES in Standardization.

[SST+05a] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, and I. Horrocks. The fuzzy description logic f-$\mathcal{SHIN}$. Proc. of the International Workshop on Uncertainty Reasoning for the Semantic Web, 2005.

[SST+05b] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan, and I. Horrocks. Fuzzy owl: Uncertainty and the semantic web. Proc. of the International Workshop on OWL: Experiences and Directions, 2005.

[SSTP05] G. Stoilos, G. Stamou, V. Tzouvaras, and J.Z. Pan. Uncertainty and ruleml rulebases:a preliminary report. In A. Adi, editor, *Proceedings of the International Conference on Rules and Rule Markup Languages for the Semantic Web*, volume 3791 of *Lecture Notes in Computer Science*, pages 199–203. Springer-Verlag, 2005.

[ST04] L. Serafini and A. Tamilin. Local Tableaux for Reasoning in Distributed Description Logics. In V. Haarslev and R. Moeller, editors, *Proc. of the International Workshop on Description Logics, DL'04*, pages 100–109, 2004.

[ST05] L. Serafini and A. Tamilin. DRAGO: Distributed reasoning architecture for the semantic web. In A. Gomez-Perez and J. Euzenat, editors, *Proc. of the Second European Semantic Web Conference (ESWC'05)*, volume 3532 of *Lecture Notes in Computer Science*, pages 361–376. Springer, May 2005.

[Sta97] C. Stace. *New Flora of the British Isles*. Cambridge University Press, 1997.

[Str01] U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence*, 14:137–166, 2001.

[Str05] U. Straccia. Towards a fuzzy description logic for the semantic web. In *Proceedings of the 2nd European Semantic Web Conference*, 2005.

[SvHSG04] H. Stuckenschmidt, F. van Harmelen, L. Serafini, and F. Giunchiglia. Unsing C-OWL for the Alignement and Merging of Medical Ontologies. In *Proc. of the KR workshop on Knowledge representation in medicine, KRMED 2004*, 2004.

[THB+93] T. G. Tutin, V. H. Heywood, N. A. Burges, D. H. Valentine, and D. M Moore(eds). *Flora Europaea*. Cambridge University Press, 1993.

[Tob01] S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, RWTH Aachen, 2001. electronically available at `http://www.bth.rwth-aachen.de/ediss/ediss.html`.

[WBH+05] Katy Wolstencroft, Andy Brass, Ian Horrocks, Phillip Lord, Ulrike Sattler, Robert Stevens, and Daniele Turi. A Little Semantic Web Goes a Long Way in Biology. In *4nd International Semantic Web Conference (ISWC2005), Galway, Ireland*, Nov 2005.

[WP05] Shenghui Wang and Jeff Z. Pan. Ontology-based Representation and Query of Colour Descriptions from Botanical Documents. In *Proc. of the 4th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE-2005)*, 2005. To appear.