D 1.4.2v3 Success Stories and Best Practices

Coordinator: Jeff Z. Pan (UAB)

Luigi Lancieri (FT), Diana Maynard (USFD), Fabien Gandon (INRIA)
Roberta Cuel (UniTn), Alain Leger (FT)

Abstract.

EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB
Deliverable 1.4.2v3 (WP1.4)
In order to make the W3C Semantic Web standards RDF and OWL more widely adopted, best practices are necessary to provide some hands-on support for developers and users of Semantic Web technologies (i.e., applications exploiting Semantic Web technologies). This deliverable analyses some best practices and related success stories of applications which make use of Semantic Web technologies. This version of the deliverable differs from version 2 in that we briefly discuss the relationship between Semantic Web best practice and ontology usability; in particular, we introduce a well understood best practice on providing scalable query answering service for OWL DL ontologies.

Document Identifier:	KWEB/2005/D1.4.2v3
Class Deliverable:	KWEB EU-IST-2004-507482
Version:	1.1
Date:	Jan 24, 2008
State:	Final
Distribution:	Public

Knowledge Web Consortium

This document is part of a research project funded by the IST Program 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 Polythechnique 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

University of Aberdeen (UAB)
Aberdeen AB24 3UE
United Kingdom
Contact person: Jeff Z. Pan
Email address: jpan@csd.abdn.ac.uk

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 writing parts of this document:

Universidad Politécnica de Madrid (UPM)
University of Manchester (UoM)

Changes

Version	Date	Author	Changes
0.7	Nov 5, 2007	Jeff Z. Pan	Initial draft based on D142v2
0.8	Nov 12, 2007	Jeff Z. Pan	Added Section 3.8
0.9	Nov 19, 2007	Jeff Z. Pan	Abstract and introduction revised
1.0	Dec 9, 2007	Jeff Z. Pan	Conclusion revised
1.1	Jan 24, 2008	Jeff Z. Pan	Revisions based on Lyndon Nixon’s comments

Executive Summary

In order to make the W3C Semantic Web standards RDF and OWL more widely adopted, best practices are necessary to provide some hands-on support for developers and users of Semantic Web technologies (i.e., applications exploiting Semantic Web technologies). The main purposes of this deliverable are to analyse some well known Semantic Web best practices, to present them in a so-called cook-book style (so as to make it easier for readers to make use of them), and to provide some example success stories related to these best practices (so as to illustrate how to make use of the presented best practices).

This version of the deliverable differs from version 2 in that we briefly discuss the relationship between Semantic Web best practice and ontology usability; in particular, we introduce a well understood best practice on providing scalable query answering service for OWL DL ontologies.

Contents

1 Introduction

The Semantic Web standards RDF and OWL have been standardised since 2004. A lot of key conferences on Semantic Web, Web 2.0, Web services, ontology based systems, etc. show a rapid growth in development of semantic technologies in industry, such as STC2006 (see D1.4.2v2) and ESTC2007. The ESTC2007 conference is the first one of its kind in Europe, it consists of about 30 use cases presentations, 6 workshops and 4 tutorials; the top 5 most represented countries are Austria, Germany, United Kingdom, Spain and France, see D1.1.5v3 [NHBL07] for more details.

Indeed, in order to make the W3C Semantic Web standards RDF and OWL more widely adopted, best practices are necessary to provide some hand-on support for developers and users of Semantic Web applications (i.e., applications exploiting Semantic Web technologies). The main purposes of this deliverable are to analyse some well known Semantic Web best practices, to present them in a so-called cook-book style (so as to make it easier for readers to make use of them), and to provide some example success stories related to these best practices (so as to illustrate how to make use of the presented best practices).

The aims of this chapter are to clarify the notions of best practices and to briefly introduce the structure of the deliverable.

1.1 Notions of best practices

1.1.1 From the viewpoint of knowledge management

One theory that has its building blocks in both organisational and technological perspectives is Knowledge Management (KM). *Knowledge management* refers to a domain of research studies and practical activities aimed at exploring and exploiting the value of knowledge generated by individuals, groups and organizations. In particular knowledge management is referred to the process of creating, codifying and disseminating knowledge within complex organisations, such as large companies, universities, and world wide organisations.

Figure 1. Adoption of Semantic Web Technology 2006

Now what [illegible] *1* [illegible] *ictice* is defined as a pattern of interlocking activities performed by a set of social actors. Since each actor performs its activity in a specific context as a consequence of its interpretations of the other's actions, a practice can be viewed as a system of actions that depend on shared expectations and interpretations. In other words, a practice is a system of activities that confirms the beliefs of interacting social agents. The term "Practice" derives from some recent epistemological approaches to knowledge that has underlined the practical nature of human activity. The practices include both the implicit and explicit knowledge, the process of learning by practicing rather than through abstract and conceptual reasoning, and the process of interiorize knowledge. Taking into consideration ethnographic studies a practice sustains a symbolic world which is functional to the cohesion of a social system. Finally from the anthropologists' perspective, a practice is also bound to complex material conditions.

A *best practice* is the best way to perform a particular system of activities in a specific context. First of all, it is "best" because, given a set of Key Performance Indicators (KPI), the selected practice is the one that maximises, among the others, the KPI set. Such selection is achieved by means of benchmarking activities, that is, the systematic comparison of practices which are aimed at achieving a similar goal, or can emerge as in an evolutionary system in which best practices emerges and are naturally selected. It is important to notice that the term "best practice" implies some contradictions. In particular the practice refers to a social process that is rooted in local contexts (that is, is a local optima), but the term best refers to some general and abstract entity which is superior to any context.

One of the most effective methods that allow practice and context transfer is the diffusion of success stories and best practices. They are good abstraction of main practices, and context constrains, therefore can be easily understood and adopted within any environment. Consider for instance that two communities (with different contexts and social constrains) recognise the need to introduce and adopt a new practice. After the introduction, the personalisation, and the adoption of this best practice, the final result will be different for both communities. In this sense, a good transfer of best practices is a negotiation among contexts aimed at generating a new boundary practice.

1.1.2 From the viewpoint of W3C

In this deliverable, we also largely rely on W3C's viewpoint of best practices. From a general point of view, the idea of collecting best practices starts from the need to have sufficient practical experience. This experience allows us to highlight consensus on positive and negative practices. Following this intuition, the W3C Semantic Web Best Practices and Deployment Working Group (SWBPD) defines the best practices as:

"A consensus-based guidance designed to facilitate Semantic Web deployment within RDF and OWL".

SWBPD aims at providing hands-on support for developers of Semantic Web applications (i.e. applications exploiting Semantic Web technologies).[1] The best practices provided by the W3C SWBPD Working Group enjoy W3C's usual consensus building culture.

[1] Currently there are quite a lot of companies that have semantic solutions R&D, see Appendix B.

1.1.3 Best practice and usability

In the context of ontology and the Semantic Web, it is getting clearer and clearer that ontology practice (including those in the area of the Semantic Web), ontology best practice and ontology usability are closely related to each other (as shown in Figure 1).

Ontology best practices come from general ontology practice; indeed, such best practices are usually results of consensus or (sometimes) well understood theoretical research. Ontology usability, on the one hand, is more general than best practice; the latter is one of the approaches to improving the form one. On the other hand, improved ontology usability could further encourage general practices of ontologies, as users of ontologies find them easier to use.

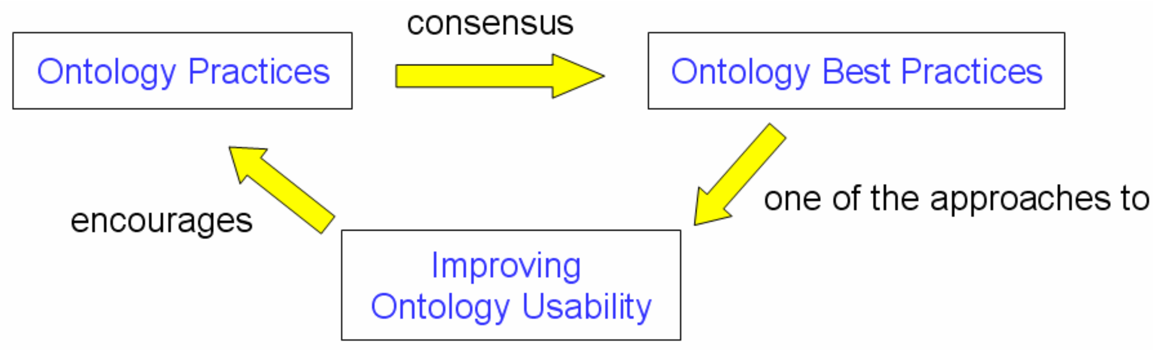

Figure 1. Ontology best practice and usability

Ontology usability, however, is a big topic and is well beyond the scope of this deliverable. In this version of the deliverable, we introduce an additional best practice (on scalable query answering service), so as to further illustrate the connection between best practice and usability. This extra best practice is related to a well know scalability problem – query answering over ontologies, which is at least as hard as entailment checking in ontologies. In fact, the decidability of query answering in OWL DL is still an open problem. Accordingly, a best practice on soundness guaranteed and scalable query answering is widely requested.

1.2 Structure of the deliverable

In order to give some tracks of thinking for future investigations, we propose an approach oriented in 3 main directions.

First of all, Chapter 2 presents a multiple choice questionnaire that integrates frequent interrogations and possible answers about researchers and practitioners' opinions on Semantic Web technologies. We adapted a methodology extracted from the field of collaborative work that enables to limit biases and optimise the statistic representativeness of answers. The motivation here is to achieve some consensus related to Semantic Web technologies and best practices. Our preliminary results show that consensus can emerge in some cases. In this version of the deliverable we will not report all the results achieved in previous work, in order to allow reviewers to easily analyse and evaluate the work done in the 2007.

Secondly and most importantly, Chapter 3 analyses some well known Semantic Web best practices, to present them in a so-called cook-book style, so as to make them easier for practitioners to check if they are related to the modeling problems they concern and to apply them if so. Most of these best practices are from SWBPD, which are (partially) contributed by researchers involved in the Knowledge Web project. This chapter is one of the main differences between this version (D142v3) and the first version (D142v1) of the deliverable. In D142v1, only some high level survey on the activities of SWBPD is provided.

Last but not least, Chapter 4 present some example success stories related to these best practices. These concrete examples aim at illustrating to practitioners how to make use of the presented best practices. Furthermore, unlike the previous version of the deliverable, there are clear connections between the success stories and the best practices presented in the previous chapter.

In short, we believe that these three complementary aspects – questionnaire on best practices and Semantic Web technologies, some well known best practices and related success stories -- will contribute to providing useful and realistic advice to the industry.

This version of the deliverable differs from version 2 in that we introduce discussions on the relationship between Semantic Web best practice and ontology usability; in particular, we present a well understood best practice on providing scalable query answering service for OWL DL ontologies. On the one hand, the new best practice makes our “cook-book” of ontology best practices more complete; on the other hand, it suggests that there exist many research problems, such as ontology usability, that are related to best practices of ontology and Semantic Web technologies.

2 Opinion Poll on Semantic Web Technologies and Best Practices

In the previous version of the deliverable the methodology and the results of the online opinion poll are deeply explained. In this chapter only some of the most important conclusions, emerged by the opinion poll, are presented.

The opinion poll is aimed at obtaining feedback on the feeling of contributors about the usefulness of best practice guidelines, as well as what these might contain. There are altogether 30 respondees of the opinion poll, The population is mainly composed of researchers (65%) including computer science (56%) and management study (7.5%). The other main part of the population (40%) is practitioners (professional developers, administrative, etc.). It should be noted that some respondents can have multiple profiles (e.g. computer science and linguist, etc). This is why we get a total percentage higher than 100%.

2.1 Some Results from the Opinion Poll

The majority (70%) think that there is a need for a clarification in practices, and developing best practice guides seems to be a reasonable approach. In this case the majority (63%) think that best practices should only consider high level advice (integration, interface, etc) and should avoid technical aspects which are too detailed. Some remarks consistent with the observation made in the previous section concerned the need for education (i.e. better practices come first from better knowledge). For 11% of the contributors, the usability of best practice guidelines is not clear and a technical tutorial is considered sufficient.

Other interesting remarks considered that best practice guidelines could be extended depending on the area of use, and in some cases could also integrate both high level and low level directives. There is also a small majority (60%) who wish to promote "labelling" through a certification authority, and who consider that basic and easily adaptable examples are better than nothing. For others (37%) it is not a good idea to implement this yet because of a lack of maturity.

An interesting divergence appears on the question related to the link between best practices and frequent practices. While 52% of the contributors think that a frequent practice should not be systematically be considered as a good practice, 45% think the contrary. This divergence induces the question of how to recognise a best practice. If we consider that a practice is based on previous uses and that expertise is based on the use of a technology, then frequent practices should be considered carefully, at least to start a recommendation repository. Alternatively we could consider that practitioners of a technology may also be influenced by bad habits coming from a "quick and dirty" adaptation of a theoretic principle. In this case frequent practices are not always good practices and "external" opinions coming from a recommendation group could be useful. Evidence for one expert is not necessarily evidence for another.

Other remarks pointed out that even if a frequent practice can provide a clue towards best practices, there is a need for more detailed technical, usage based advice or examples in order to be pedagogically useful.

2.1.1 Ontology and the real world

This section relates to the level of realism that ontologies should achieve. The question could be formulated in the following way: do we need practical concepts and tools which are easy to use if they only reflect poorly the real world, or should we instead promote precision in knowledge representation at the risk of introducing complexity?

Regarding the involvement of philosophers in Semantic Web, only 22% think that this is not a good idea (lack of pragmatism, difficult to manage, etc.) whereas 18% are clearly favourable. Actually the majority (60%) is mostly undecided and thinks that it should dependent on the context and application. The ratio is quite similar regarding the involvement of logicians. On the other hand, it seems that the help of linguists is a little more appreciated, since only 4% of the contributors think that a linguist would not be useful, whereas 33% are favourable and 70% think that it depends on the context and the application.

Uncertainty is linked to our perception of reality, and it is well known that our natural cognitive processes are mainly based on probabilistic reasoning. It could be interesting to ask whether uncertainty and probability need to be taken into account in the Semantic Web. The majority (67%) of the contributors answer yes to this question. The comments also clearly show that the Semantic Web is not mature enough to take into account these aspects.

2.1.2 Building ontology

Following from the previous question, we consider here practical aspects of ontology building.

Several respondents pointed out that RDF is very limited and cannot alone ensure the needs of the Semantic Web. Only 37% think that RDF alone could be enough, whereas 70% think that RDF and OWL are enough. 47% of the contributors prefer the use of a limited version of OWL (Lite, DL) instead of OWL Full. 37% think that embedding RDF in another technology (HTML, RSS, etc) should be recommended, whereas 26% recommend avoiding it (see details of the technical concerns in the questionnaire).

The majority (56%) of the contributors think that a domain oriented ontology (fit to the problem to be solved) should be recommended, whereas 30% think that a general ontology (a portable ontology usable in a maximum number of domains) is preferable, and 33% think that no rules should be recommended in this matter. Comments pointed out that the best way is probably to promote a domain oriented ontology linked to a general ontology.

The majority (78%) of the contributors think that the quantity of concepts used in a Semantic Web application should remain free since it depends on the application. About 10% think that there is a need for a maximum limit in order to reduce the complexity, possible inconsistency or to maintain good performance within the application.

For the majority (80%), the security aspects of an ontology mainly depend on the needs and context and it is difficult to be formalise these in strict rules.

The majority (67%) think that we need to recommend the use of ontology building from text (15% do not agree), such as tools for cleaning ontologies (59%) and consistency verification tools (74% in all cases, 19% only in complex cases).

Most contributors (41%) employ an ontology using only one natural language, whereas 26% use 2 or more. Regarding the representation language, 19% use one language whereas 19% use two and 19% use more than two. 40% of the contributors use synonyms for keywords, while 22 % do not.

2.1.3 Availability and reusability of ontologies

In order to improve the reusability of ontologies, we may wonder how to manage their availability. This includes preliminary considerations like persistency (i.e. building ontologies to be reusable, live for a long time, etc) but also the strategy of institutions (whether an ontology is freely available, etc.).

The majority of contributors (85%) think that an ontology is supposed to be persistent for a long time and can be used for several generations of applications. In such a case, a dedicated maintenance effort is necessary. Respondents also pointed out that this could depend on the context and that in some cases an ontology could have a limited time to live.

Strangely, only 18% of the contributors are sure that the Semantic Web will reach a high level of reusability, whereas 30% think that reusability will be low and 48% hope that the reusability will be high but that it is not clear that this will be the case. One respondent pointed out the need for popularisation of the ontology "model" (well modularised, easy to use, etc.).

Regarding the reuse of existing conceptualisations (database schemas, text, etc.), 48% of the contributors think that this should be promoted whereas 4% think the contrary and 44% think that it depends on the application. As suggested by some remarks, it is possible that the conceptualisation that the Semantic Web will ultimately be based on is not yet known. In such cases of conceptualization evolution, reuse of existing conceptualisations is certainly a need.

A majority is favourable to a mapping between new and existing ontologies (as a priority, 48%; if there is a need, 37%). The results show that reusability is a real concern within the Semantic Web community. Thus, 85% are considering adapting or extending an available ontology to their projects, whereas 37% prefer to develop their own ontology. The big discussion and opposition between specificity / optimality and openness / reusability appears again, considering that 52% think that an ontology would be more efficient if developed by an individual organization to fit their specific needs, whereas 37% think that this would be more efficient if done by a public institution in order to ensure authority, consensus, and trust. 30% do not have a clear idea on this subject.

Regarding availability, 37% think that ontologies should be available publicly, free of restrictions, whereas 48% think that it depends on the applications and that they could in certain cases be released under license.

2.1.4 Using ontologies

This section is intended to give a feed back on the main uses of ontologies. The idea is to evaluate the level of applications where knowledge formalism is involved in machine to machine cooperation.

The results show that ontologies are used in a wide variety of applications; some (67%) are still mainly related to human-machine interaction (help with information search, browsing, etc.) whereas 63% are mainly inter-process related. The use of ontologies in e-business is 44%, but seems very promising as well as information disclosure and information integration. At the moment, security concerns do not seem to be a priority and few are taken into account in applications.

3 Best Practices of Semantic Web Technologies

3.1 Introduction

This chapter presents and analyses some well known Semantic Web best practices. Most of the best practices presented in this chapter are (partially) contributed by Knowledge Web researchers in the W3C Semantic Web Best Practice and Deployment (SWBPD) Working Group.

In particular, we will present the best practice in a so-called cook-book style, covering the following aspects:

- the problem(s);
- solutions:
    - ingredients;
    - required materials (ontology expressive power);
    - examples;
- tips (discussions on, e.g., pros and cons) of the best practices.

3.2 Representing Quality Values[2]

3.2.1 The problem

How to represent values of qualities, such as size, severity, texture and rank, in ontologies?

3.2.2 Solution 1: Values as subclasses partitioning a quality

3.2.2.1 Ingredients

- A quality is represented as a class
- Typical value sets of a quality are represented as sub-classes of the quality class

3.2.2.2 Required expressive powers

- Class equivalent axioms
- unionOf class constructor
- existential restrictions

[2] This is based on the SWBPD working draft *Representing Specified Values in OWL: "value partitions" and "value sets"* edited by Alan Rector (University of Manchester).

3.2.2.3 Example: John is in good health

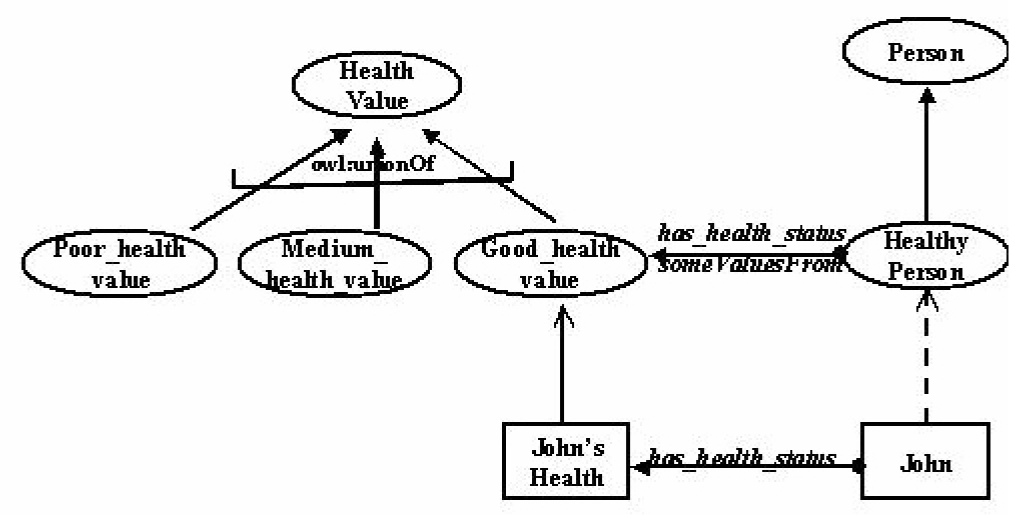

Figure 1 Solution 1: Values as subclasses partitioning a quality (from the SWBPD working draft *Representing Specified Values in OWL: "value partitions" and "value sets"*)

To say that "John is in good health" is to say that his health is inside the good_health_values partition of the Health_value quality. See Figure 1 and Ontology 1 (in OWL abstract syntax) for details.

Ontology 1:

Class (HealthValue complete
        unionOf (Poor_health_value Medium_health_value Good_health_value))
Class (HealthPerson complete intersectionOf (Person
        restriction (has_health_status someValuesFrom (Good_health_value))))
Individual (John type (Person) value (has_health_status (John'sHealth)))
Individual (John'sHealth type (Good_health_value))

3.2.3 Solution 2: Values as individuals whose enumeration is equivalent to the quality

3.2.3.1 Ingredients

- A quality is represented as a class
- Typical values of a quality are represented as instances of the quality class

3.2.3.2 Required expressive powers

- Class equivalent axioms
- The unionOf class constructor
- existential restrictions

- The nominal (oneOf) class constructor

3.2.3.3 Example: John is in good health

To say that "John is in good health" is to say that his health is inside the good_health_values partition of the Health_value quality. See Figure 2 and Ontology 2 (in OWL abstract syntax) for details.

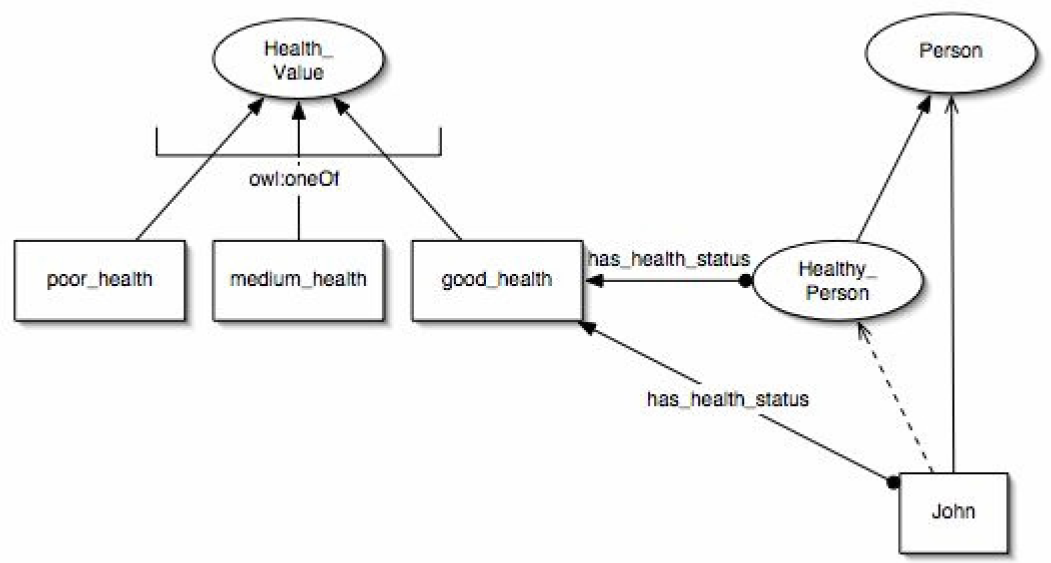

Figure 2 Solution 2: Values as individuals whose enumeration is equivalent to the quality (from the SWBPD working draft Representing Specified Values in OWL: "value partitions" and "value sets")

Ontology 2:

```
Class (HealthValue complete
        unionOf (oneOf (poor_health_value) oneOf  (medium_health_value)
                oneOf (good_health_value)))
Class (HealthPerson complete intersectionOf (Person
        restriction (has_health_status someValuesFrom (oneOf (good_health_value)))))
Individual (John  type (Person) value (has_health_status (good_health_value)))
```

3.2.4 Tips

- Both solutions correctly classify John as an instance of the HealthyPerson class.
- The advantage of the first solution is that it does not require the use of nominals.
- The advantage of the second solution is that values are represented as individuals rather than classes – many people think this is more intuitive.

- Both solutions are not precise enough to capture the value constraints, such as classifying John is an adult based on his age. See Section 3.5 and Section 4.2 for more detailed discussions on using datatypes to represent qualities.

3.3 Representing Relations with Arbitrary Arities[3]

3.3.1 The problem

How to represent (N-ary) relations among more than two individuals in RDF and OWL (which support only binary relations)?

3.3.2 Solution 1: Distinguishing the originator individual

3.3.2.1 Ingredients

- Distinguishing the originator individual
- N-ary relations are represented as a class plus N binary relations

3.3.2.2 Required expressive powers

- The intersectionOf class constructor
- Existential restrictions
- Functional property axioms

3.3.2.3 Example: Steve has temperature, which is high, but falling

To say that "Steve has temperature, which is high, but falling" is to say that Steve relates via the property has_temperature a complex object representing different facts about his temperature. See Figure 3 and Ontology 3 (in OWL abstract syntax) for details.

[3] This is based on the SWBPD working *draft Defining N-ary Relations on the Semantic Web: Use With Individuals* edited by Natasha Noy (Stanford University) and Alan Rector (University of Manchester).

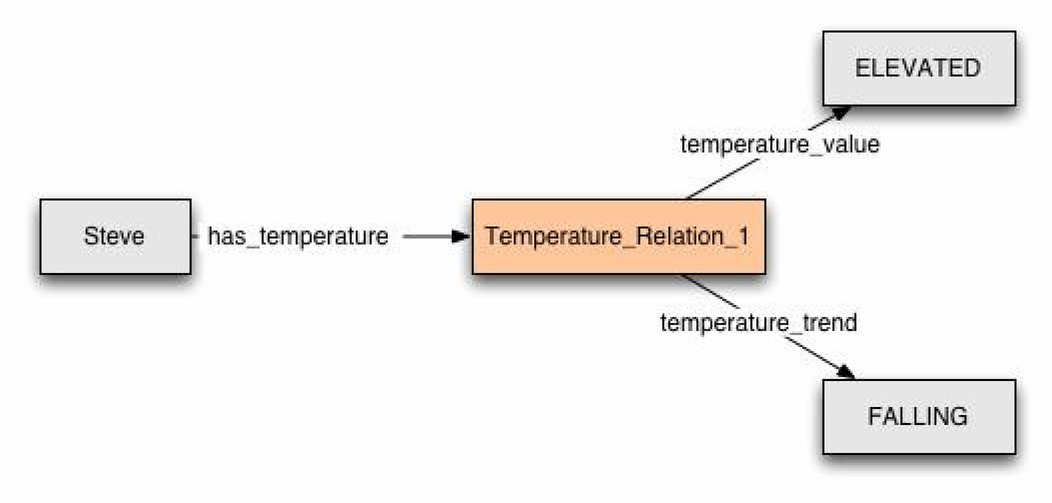

Figure 3 Solution 1: Distinguishing the originator individual (from the SWBPD working *draft Defining N-ary Relations on the Semantic Web: Use With Individuals*)

Ontology 3:

```
ObjectProperty (has_temperature Functional)
ObjectProperty (temperature_value Functional)
ObjectProperty (temperature_trend Functional)
Individual (Steve value (has_temperature
        Individual (type (intersectionOf
        (restriction(temperature_value someValuesFrom (Elevated))
        restriction (temperature_trend someValuesFrom (Falling))))))
```

3.3.3 Solution 2: No originator individual

3.3.3.1 Ingredients

N-ary relations are represented as a class plus N binary relations.

3.3.3.2 Required expressive powers

Functional property axioms.

3.3.3.3 Example: John buys a "Lenny the Lion" book from books.example.com for $15 as a birthday gift.

To say that "John buys a ‘Lenny the Lion’ book from books.example.com for $15 as a birthday gift", we introduce the Purchase_1 object (as an instance of the N-ary relation class) that relates other individuals (such as John). See Figure 4 and Ontology 4 (in OWL abstract syntax) for details.

Ontology 4

```
ObjectProperty (buyer Functional)
ObjectProperty (seller Functional)
ObjectProperty (object)
ObjectProperty (purpose)
ObjectProperty (amount Functional)

Individual (Purchase_1 type (Purchase)
        value (buyer Individual (John))
        value (seller Individual (books.example.com))
        value (object Individual (Lenny_the_Lion))
        value (purpose Individual (birthday_gift))
        value (amount Individual ($15)))

/* Note that we can better represent $15 by using an object property unit and a datatype
property value.*/
```

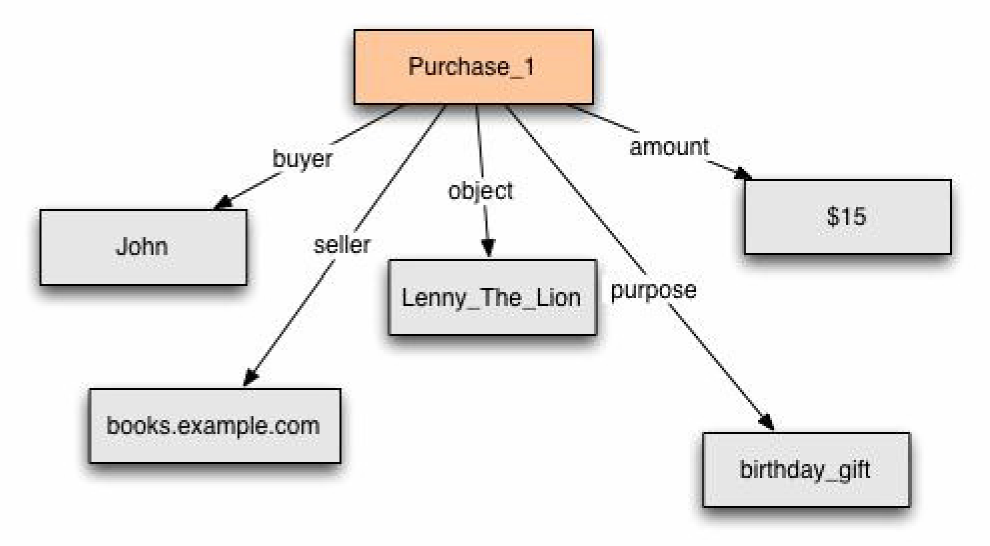

Figure 4 Solution 2: No originator individual (from the SWBPD working draft Defining N-ary Relations on the Semantic Web: Use With Individuals)

3.3.4 Tips

- The choice between the two solutions is subjective.
- Solution 1 usually requires the use of anonymous individuals, using existential restrictions seems to be more convenient.
- Functional property axioms are essential in both solutions. Otherwise, we might have more than one temperature_value for Steve and more than one buyer for Purchase_1.

3.4 Qualified cardinality restrictions (QCRs)[4]

3.4.1 The problem

How to represent qualified cardinality restrictions (QCRs) in OWL (which support only unqualified cardinality restrictions)?

3.4.2 Solution 1: Existential restrictions

3.4.2.1 Ingredients

“At least one” QCRs can be represented as existential restrictions.

3.4.2.2 Required expressive powers

Existential restrictions.

3.4.2.3 Example: Person who has at least one parent who is a British citizen

To say that "Person who has at least one parent who is a British citizen", we can use the following OWL class axiom

```
Class(Person_with_British_parent partial
     intersectionOf(Person
         restriction(has_parent someValuesFrom(British_Citizen))))
```

3.4.3 Solution 2: Sub-property and range property axioms

3.4.3.1 Ingredients

Introducing a sub-property of the primary property and then to introduce an unqualified cardinality restriction on that sub-property.

3.4.3.2 Required expressive powers

- Unqualified cardinality restriction
- Sub-property axioms
- Property range axioms

[4] This is based on the SWBPD working draft *Qualified cardinality restrictions (QCRs)* edited by Alan Rector (University of Manchester) and Guus Schreiber (Free University of Amsterdam).

3.4.3.3 Example: The normal hand has exactly five fingers of which one is a thumb

We can represent “the normal hand has exactly five fingers of which one is a thumb” with the following OWL class axioms

```
Class(Finger partial Body_part)
Class(Thumb partial Finger)

ObjectProperty(has_part                      range(Body_part))
ObjectProperty(has_finger super(has_part)    range(Finger))
ObjectProperty(has_thumb  super(has_finger) range(Thumb))

Class(Normal_hand partial
    intersectionOf(
      restriction (has_finger cardinality(5))
      restriction (has_thumb cardinality(1))))
```

3.4.4 Tips

- Solution 1 only applies on “at least one” QCRs.
- Solution 2 introduces (unnecessary) global range constraints, while QCRs are simply local constraints.
- Solution 2 is a “work-around”: (1) it is not enough to capture the complete semantics (see also Jeff Z. Pan’s comments on this http://lists.w3.org/Archives/Public/public-swbp-wg/2005Dec/0125.html); (2) the “work-around” is fine if there exist no inappropriate axioms about the related properties and classes. Two rules of thumb are (1) not to use the primary property directly but always use the sub-properties in cardinality restrictions and (2) to make sure the sub-properties have different ranges.
- In Section 4.1, we present a success story related to this best practice.

3.5 XML Schema User Defined Datatypes in RDF and OWL[5]

3.5.1 The problem 1

What is the relationship between the value spaces of the various XML Schema built-in simple types when used within RDF and OWL? Or in other words, when do two literals, which are written down differently, refer to the same value?

[5] This is based on theW3C SWBPD Note *XML Schema Datatypes in RDF and OWL* edited by Jeremy Carroll (HP Lab) and Jeff Z. Pan (University of Aberdeen).

3.5.2 Solution of problem 1

3.5.2.1 Ingredients

- All primitive XML Schema Datatypes are treated as having disjoint value spaces.
- Meaningful mapping among values from different primitive datatypes could be enabled by using the value approximate map and approximate equality.
- A *value approximate map* mapsTo is a partial mapping from typed literals to typed literals.
- Given a datatype map D and a value approximate map mapsTo, the approximate equality aeq is defined as follows (NB: L2S refers to lexical to value mapping):
    - aeq("s1"^^u1, "s2"^^u2)=true if L2S(D(u1))(s1) = L2S(D(u2))(s2) or if mapsTo("s1"^^u1)="s3"^^u2 and L2S(D(u2))(s3) = L2S(D(u2))(s2);
    - aeq("s1"^^u1, "s2"^^u2)=false otherwise.

3.5.2.2 Examples

- `"15"^^xsd:byte` and `"15.0"^^xsd:decimal` both denote the same value, fifteen. This follows because `xsd:byte` has primitive base datatype `xsd:decimal`. Therefore, Individual (Jane value (age `"15"^^xsd:byte))` entails Individual (Jane value (age `"15.0"^^xsd:decimal))`.
- "1.3"^^xsd:decimal is different from "1.3"^^xsd:float, as xsd:decimal and xsd:float are two different primitive datatypes.
- Given the value approximate mapping mapsTo("1.3"^^xsd:decimal)= "1.3"^^xsd:float, we have the following approximate equality aeq("1.3"^^xsd:decimal, "1.3"^^xsd:float)=true.
- Given the above approximate mapping and the following two individual axioms: Individual (car1 value (enginesizeInLitre "1.3"^^xsd:decimal) and Individual (car2 value (enginesizeInLitre "1.3"^^xsd:float), the following SPARQL query

```
SELECT  ?size
WHERE   { eg:car eg:engineSizeInLitres ?size .
        FILTER (?size = 1.3) . }
```

returns both car1 and car2.

3.5.3 The problem 2

How to integrate XML Schema user-defined datatypes with OWL DL?

3.5.4 Solution of problem 2

3.5.4.1 Ingredients

- In order to support XML Schema user-defined datatypes with OWL DL (in general not just OWL DL but a large family of decidable, including very expressive, Description Logics), one needs to extend OWL datatyping to unary datatype groups. Intuitively speaking,
- A combined DL is decidable if the unary datatype group is conforming. A *conforming* unary datatype group is equipped with a decision procedure for the satisfiability problem of finite conjunctions over supported datatypes.
- In a unary datatype group, OWL data ranges are extended to datatype expressions so as to represent user defined datatypes. Let G be a unary datatype group, the set unary datatype expressions for G, abbreviated Dexp(G), is inductively defined as follows:
    - let u be a datatype URI reference, u ∈ DPexp(G);
    - let u be a datatype URI reference, its (relativised) negation not(u) ∈ DPexp(G);
    - let y_1, ..., y_n be literals, the enumerated datatype oneOf(y_1, ..., y_n) ∈ DPexp(G);
    - for any p,q ∈ DPexp(G), their conjunction and(p,q) ∈ DPexp(G);
    - for any p,q ∈ DPexp(G), their disjunction or(p,q) ∈ DPexp(G).

3.5.4.2 Required expressive powers

- Unary datatype groups
- Unary datatype expressions

3.5.4.3 Examples

As a further example, we may wish to talk about ages of adults in years, where an adult is over 18. This can be described as a restriction on the `xsd:integer` datatype.

```
<xsd:simpleType name="adultAge">
  <xsd:restriction base="nonNegativeInteger">
   <xsd:minInclusive value="18">
  </xsd:restriction>
</xsd:simpleType>
```

This datatype can be represented as the following unary datatype expression:

```
and (xsd:nonNegativeInteger, xsdx: minInclusive18).
```

We can use this unary datatype expressive to define the Adult class:

```
DatatypeProperty (age Functional)
Class (Adult complete intersectionOf (Person
        restriction (age someValuesFrom
                and (xsd:nonNegativeInteger, xsdx: minInclusive18))))
```

3.5.5 Tips

- [PaHo2005] shows that we can combine any decidable DL (including SHOIN, the underpinning of OWL DL) that provides the conjunction and bottom class constructors with a conforming (unary) datatype group and the combined DL is still decidable.
- Being able to use user-defined datatypes in ontologies, we can directly represent quality values (see Section 3.2) as datatyped values. An example and more detailed discussions are presented in Section 4.2.

3.6 Representing Object-Oriented Classes and Attributes[6]

3.6.1 The problem 1

How to represent object-oriented classes which do not share instances?

3.6.2 Solution of problem 1

3.6.2.1 Ingredients

Explicitly assert that all the named classes are disjoint.

3.6.2.2 Required expressive powers

Disjoint class axioms.

3.6.2.3 Examples

Suppose there are only two classes in the ontology, which are Product and Customer, we can assert that they are disjoint.

Class (Product)
Class (Customer)
DisjointClasses (Product Customer)

3.6.3 The problem 2

How to represent object-oriented attributes which (1) are local to corresponding classes and (2) have ranges that are used for type checking?

[6] This is related to the W3C SWBPD Note *A Semantic Web Primer for Object-Oriented Software Developers* contributed by Holger Knublauch (University of Manchester), Daniel Oberle (Universität Karlsruhe), Phil Tetlow (IBM), Evan Wallace (National Institute of Standards and Technology) and Jeff Z. Pan (University of Aberdeen).

3.6.4 Solution of problem 2

3.6.4.1 Ingredients

- Representing locality of attributes by using existential restrictions.
- Representing type checking for property ranges with value restrictions

3.6.4.2 Required expressive powers

- Existential restrictions
- Value restrictions
- The intersectionOf and complementOf class constructors

3.6.4.3 Examples

To say the class customer has an attribute email, we can use the following axiom.

```
Class (Customer partial restriction (address someValuesFrom (xsd:string)))
```

That is, for each customer, there exist a string which is his/her address.

To set the range of email as xsd:string for type checking, we can use the following axiom.

```
SubClassOf (
        intersectionOf (Customer
                complementOf (Restriction (email allValuesFrom xsd:string))
        owl:Bottom)
```

3.6.5 Tips

One of the most convincing advantages of using Semantic Web technologies to support object-oriented modelling is that the domain model can be shared online and can be dynamically maintained in run time.

3.7 Scalability of Natural Language Tasks

3.7.1 The Problem

Inherent difficulties in language processing tasks (e.g. incompleteness, language change, ambiguity, etc.) make it very difficult to scale HLT applications from research prototypes to real world applications.

3.7.2 Solution 1

Restrict the scope of applications to smaller domain-specific, tightly focused tasks which can be performed automatically with high accuracy.

3.7.2.1 Example of Solution 1

Semantic annotation systems which attempt to cover any domain, e.g. to annotate the whole internet, are doomed to a low level of accuracy if they are to be fully automatic. Dividing the problem into bite-sized chunks, such as having one system for news texts, another for finance texts, etc. results in a series of smaller, related systems with high accuracy.

3.7.3 Solution 2

Development of semi-automatic systems that rely on a certain amount of human assistance, using manual training to teach the system, manual intervention to check problematic issues, or manual post-editing to refine system output, or a combination of the above

3.7.3.1 Example of Solution 2

Semantic annotation systems which make use of mixed-initiative learning, whereby the user begins annotating the data manually, providing input to the system which gradually learns, and finally takes over more and more of the task until it is running fully automatically (but enabling manual post-editing as necessary).

3.8 Semantic Approximation of OWL DL Ontologies

3.8.1 The Problem

How to provide efficient querying answering service for expressive Description Logics has been an important open problem in KR. In fact, whether query answering in OWL DL is decidable is still an open problem. Syntactic approximations can guarantee neither soundness nor completeness.

3.8.2 Example of Limitation of syntactic approximations

Syntactic approximations are approximations that translate axioms in the source ontologies into axioms in a weaker ontology language. Let us consider an OWL DL source ontology O = {$\exists_{\geq 50}$ hasGoodFriend $\sqsubseteq$ Popular (anyone with at least 50 good friends is popular), hasGoodFriend $\sqsubseteq$ hasFriend (if A has good friend B, then A has friend B), hasGoodFriend(john,kate) (john has a good friend kate)}.

If we consider to approximate it with an ALC ontology using syntactic approximation, we have to give up the expressive power of number restrictions and property subsumptions. Accordingly, the above OWL DL ontology O can be syntactically approximated into the following ALC ontology O' = {{$\exists_{\geq 1}$ hasGoodFriend $\sqsubseteq$ Popular (anyone with at least a good friend is popular), hasGoodFriend (john,kate) (john has a good friend kate)}.

It is obvious that reasoning in O’ is neither sound nor complete.

- Unsoundness: O’ entails P(john) (john is popular) but O does not.
- Incompleteness: O entails hasF(john,kate) (john has a friend kate) but O’ does not.

3.8.3 Solution

Restrict the scope of applications to smaller domain-specific, tightly focused tasks which can be performed automatically with high accuracy. In order to guarantee soundness, the solution is to semantically approximate an source OWL ontology Os with its least upper bound Ot, i.e., the set of entailments that are representable in a simpler target language.

For query answering, we choose DL-Lite as the target language, in which the entailment sets are always finite. Semantic approximation guarantee soundness for query answering; for database style queries (in which variables appearing in the head of the queries also appearing in their bodies), it also guarantees completeness. [PaTh07]. It is a general approach and works not only on OWL DL.

Semantic approximation has been implemented in the ONTOSEARCH2 system (www.ontosearch.org). ONTOSEARCH2 outperforms existing SHIQ query engines (such as Pellet and KAON2) and syntactic approximation systems such as DLDB (see the following figure on evaluation with the LUBM – (0,50) means 50 universities in which case the ontology has about 7 million individuals).

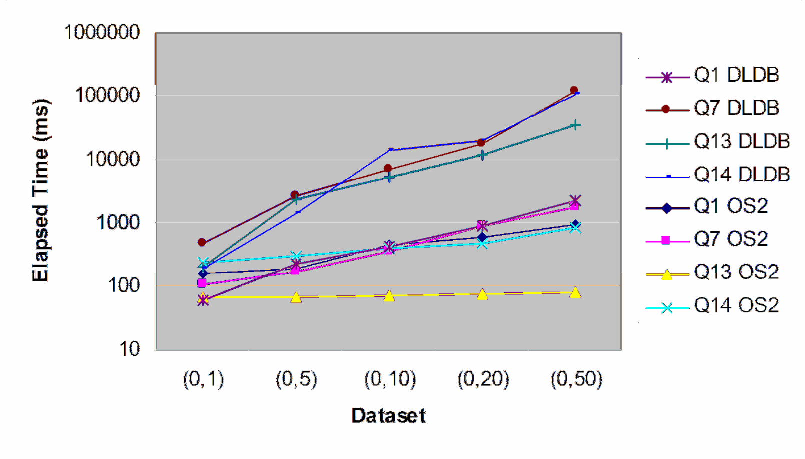

Figure 5 DLDB vs OS2

From the above figure, the performance of ONTOSEARCH2 and DLDB are comparable when the ontologies are not so huge (such as 1-5 universities, which corresponds to about 100-700 thousand individuals) Note that in the LUBM benchmark, all the queries are database-style queries, so ONTOSEARCH2 provides sound and complete results, while DLDB is a syntactic approximation system.

In short, the main advantage of semantic approximation is that users can choose expressive ontology languages (such as OWL DL or OWL 1.1) to represent their domain knowledge, while they can still enjoy scalable reasoning services provided only by lightweight ontology languages, as well as soundness guaranteed results.

3.8.4 Tips

It is easy to distinguish database style queries from the rest： all their variables appear in the head.

4 Example of Success Stories

This chapter presents some example success stories related to the best practices presented in the previous chapter.

4.1 Qualified Cardinality Restrictions – Classifying Protein Sub-Families

This success story is based on the work presented in [WBHL*05], and is related to the best practice presented in Section 3.4.

4.1.1 Motivation

Proteins classification is a central process in understanding the molecular biology. Such classification is based on functional domains of proteins. Many proteins are assemblies of domains. Each domain has a separate function. Domain compositions decide protein functions. The recognition of domain composition in a fine-grain level requires analysis of bio-informaticians. Now the challenge is to capture understanding of bio-informaticians and apply systematically within computer applications.

4.1.2 Key ideas

Wolstencroft and colleagues [WBHL*05] use an OWL-DL ontology to represent the expert knowledge and use Instance Store (a DL-based reasoning system) to perform classification. One of the main problems is to capture expert knowledge like the following:

"If a protein Y contains at least n1 and at most n2 p-domains of type X1, and at least n3 and at most n4 p-domains of type X2, then Y belongs to family Z."

To capture this knowledge, we need to use qualified cardinality restrictions (QCRs) which is not provided in OWL. To solve the problem, we can use the work-around provided in Section 3.4 to define the class Z as follows.

```
Class (Z complete intersectionOf
        restriction (hasX1-p-domain minCardinality(n1) maxCardinality(n2))
        restriction (hasX2-p-domain minCardinality(n3) maxCardinality(n4))).
```

Note that instead of using the has-p-domain property, we use different properties for different domain types, and that we should have one property (e.g. hasX1-p-domain) for each domain type (e.g. X1), in order to ensure the completeness.

The results of this work are three-fold. Firstly, the automated classification of the human protein phosphatases performed as well as the manual classification by phosphatase experts. Secondly, the automated classification discovered two proteins for which no appropriate family was available. This discovery led to a modification of the ontology and thus of the expert knowledge on proteins. Thirdly, the automated classification discovered some mis-classified A.fumigatus phosphatases, and revealed large differences from the human phosphatases.

4.1.3 Discussion

Besides finding new protein families that are of interest to biologists, this work has shown that automated classification can indeed compete with manual classification, and is sometimes even superior. This approach combines the advantages of speed of the automated methods and accuracy of human expert classification, the latter being due to the fact the we can capture the expert knowledge in an OWL ontology. The combination of the two, namely speed and expert knowledge, provides a quick and efficient method for classifying proteins on a genomic scale.

4.2 XML Schema User Defined Datatypes – Integrating and Querying Leave Shapes

This success story is based on work reported in [WaPa05] and [WaPa06], and is related to the best practices presented in Sections 3.2 and 3.5.

4.2.1 Motivation

The processing of words and phrases for continuous quantities raises important issues in the treatment of semantics. The problem becomes particularly focused when we consider its computational aspects. The demand that a semantics be computational means that, not only is the interpretation of phrases (efficiently) computable, but also the extent that two denotations are equal, equivalent, close, or overlap may itself be (efficiently) computed. As one of the premier descriptive sciences, botany offers a wealth of material on which to for evaluation. Now the challenge is how to use ontologies to facilitate integrating and querying information on parallel colour and leaf shape descriptions from botanical documents.

4.2.2 Key ideas

We consider flower colour and leaf shape as examples to illustrate how to represent descriptions of these quantities in an ontology system, using the OWL-Eu ontology language [PaHo05]. We have devised a plant ontology O, which contains Colour and LeafShape as primitive classes. Other primitive classes in O include

Class(Species), Class(Flower), Class(Colour), Class(Leaf), Class(LeafShape);

important object properties in O include

ObjectProperty(hasPart), ObjectProperty(hasColour), ObjectProperty(hasShape).

In order to describe precisely values of continuous properties, such as hasHue and hasSaturation, we need to use data values, rather than individuals. Therefore, we consider the following datatype properties

hasHue, hasSaturation, hasLightness,
hasLengthWidthRatio, hasBroadestPosition, hasApexAngle and hasBaseAngle,

which are all functional properties. Each datatype property and its range is also defined, for example:

DatatypeProperty (hasBaseAngle Functional range(and(≥0, ≤180))).

Concrete colours and leaf shapes are defined based on the above primitive classes and properties, where datatype expressions are used to express the semantic regions. For example, the colour ‘purple’ and the shape ‘ovate’ is defined as the following OWL-Eu classes (using unary datatype expressions such as and(≥78, ≤88):

```
Class (Purple complete intersectionOf (Colour
        restriction (hasHue someValuesFrom (and(≥78, ≤88))
        restriction (hasSaturation someValuesFrom (and(≥45, ≤55))
        restriction (hasLightness someValuesFrom (and(≥20, ≤30))))
```

```
Class (Ovate complete intersectinOf (LeafShape
        restriction (hasLengthWidthRatio someValuesFrom (and(≥15, ≤18))
        restriction (hasBroadestPosition someValuesFrom (and(≥39, ≤43))
        restriction (hasApexAngle someValuesFrom (and(≥41, ≤50))
        restriction (hasBaseAngle someValuesFrom (and(≥59, ≤73)).
```

In the end, a species with ‘purple’ flower and ‘ovate’ leaf shape can be represented as an OWL-Eu class

```
Class (SpeciesA complete intersectionOf (Speacies
        restriction (hasLeaf someValuesFrom (LeafA))
        restriction (hasFlower someValuesFrom (FlowerA))))
Class (LeafA complete intersectionOf (Leaf
        restriction (hasShape someValuesFrom (Ovate))))
Class (FlowerA complete intersectionOf (Flower
        restriction (hasColour someValuesFrom (Purple))))
```

Similarly, species with complex flower colour and leaf shapes are also defined as OWL-Eu classes, with their flower colour and leaf shape represented as OWL-Eu classes.

It is important to note that being able to use user-defined datatypes (unary datatype expressions) to represent ranges used in colour and shape descriptions enable us to precisely capture mathematical semantic model about colour and shape descriptions.

4.2.3 Discussion

Ontological representations of colour and shape descriptions, together with appropriate distance functions for each property, enable us to integrate parallel descriptions and to carry out species identification queries based on their flower colour and/or leaf shapes as we now describe. Wang and Pan [WaPa06] further did some evaluations on species identification queries, and this approach outperforms the keyword-based method; this is because the former takes the real semantic of shape descriptions into account and thus make semantic similarity measurement possible, while the latter simply checks the word matching of the descriptions.

4.3 Semantic Annotation -- SWAN

This success story is related to the best practices presented in Sections 3.7.

4.3.1 Motivation

There is currently much work in the area of semi- and fully automatic semantic annotation, but until now there has always been a tradeoff between performance and scalability. While performance is clearly important, the Semantic Web will never be a reality unless applications are fully scalable and can cope with enormous volumes of data. Systems that are designed for massive annotation are generally automatic, non-specific and do not have a high level of performance. Smaller systems may perform well but are not scalable to large amounts of data.

4.3.2 Key ideas

SWAN (Semantic Web ANnotator) is a system designed to perform large-scale ontology-based information extraction for the Semantic Web, annotating vast amounts of documents from the web with semantic information (inferred metadata). The annotation process can be viewed as a chain of logical components, starting with the crawling of documents from the web and ending with the user of the platform receiving a semantic response to a query. The system is based largely on KIM [Pop04a], which provides indexing, disambiguation and storage components, as well as some of the interface components.

SWAN contains two focused crawler versions: an HTML crawler which directly accesses web pages according to a defined scope, and an RSS crawler which uses the syndication mechanism of RSS 1.0 newsfeeds. The RSS crawler has the advantage of being already domain-specific and therefore more likely to return relevant documents, and some "free" (explicit) metadata such as author name and publication date. The web pages found are then passed to the IE component, which consists of a set of processing resources implemented using GATE [Cun02b]. This pipeline of resources performs preprocessing tasks such as tokenisation and sentence splitting, followed by high-level pattern matching and coreference resolution, and results in a set of semantic annotations linking the text with concepts from an ontology. The disambiguation component then performs 2 tasks: first, it co-refers different mentions of the same instance at the document level, and second, it continuously checks if new instances found are identical to previously found entities in other documents (and thus already contained in the Knowledge Repository). Finally, the results are stored in various databases. Entities, relations and their properties are stored in an RDF Knowledge Repository, using Sesame[7]. An index relating the entities to their source documents is stored in a Document Store, implemented on top of Lucene[8]. The annotations themselves are stored in an Annotation Store implemented as a relational database.

SWAN allows access to its data for humans via a web-based UI, using an ordinary web browser, which allows the user to enter queries, e.g. "Who are the CEOs of companies in Ireland?", and to access the results via a web page. They can also pose queries directly in a formal query language such as RQL or SeRQL, and access the results as RDF statements about the entities matching the query. The system is designed to work on specific domains, because the accuracy is vastly improved in this way. However, it is also deliberately designed to be scalable, and new domains are being continuously added.

4.3.3 Discussion

SWAN has been evaluated in a number of ways. The problem of scalability with respect to crawling and annotation is dealt with by organising the components in a cluster architecture of 4 annotator machines responsible for the extraction process. A document queueing system divides the load between the 4 machines. The crawler places each downloaded document on top of the queue, and each annotator in turn takes a document from the queue and processes it. An upper limit is set for the queue size to prevent overload -- if this limit is reached then the crawler halts temporarily. The number of machines could of course be increased, should the need arise. A distributed architecture has not been implemented for storage, but the current architecture appears to scale well in tests so far. SWAN deals with the performance aspect of scalability by focusing on specific domains rather than attempting to cover all topics with one single application, as described in Section 3.7.

[7] http://www.openrdf.org

[8] http://lucene.apache.org

5 Discussion and Outlook

In this report, we have analysed some well known Semantic Web best practices, most of which are (partially) contributed by Knowledge Web researchers in the W3C Semantic Web Best Practice and Deployment Working Group (SWBPD). In particular, we provide a cook-book style of presentations for the SWBPD-related best practices, highlighting the problem(s), solution(s) and tips about them. It should be noted that the deliverable is not intended to be a replacement of all the related W3C technical reports: on the one hand, it aims at providing a compact version so as to make it easier for readers to get the main points; on the other hand, it suggests that there exist many research problems, such as ontology usability, that are related to best practices of ontology and Semantic Web technologies. Indeed, ontology usability plays an important role on encouraging the generation of such best practices; e.g., solutions (such as semantic approximation) of some of the ontology usability issue (such as scalable query answering service for OWL DL) are useful best practices.

In order to make the W3C Semantic Web standards RDF and OWL more widely adopted, best practices are necessary to provide some hand-on support for developers and users of Semantic Web applications (i.e., applications exploiting Semantic Web technologies). To this end, the SWBPD Working Group has done a nice job. Although it was closed in May 2006, there are three subsequent groups started right after its ending: (1) W3C Multimedia Semantics Incubator Group, (2) W3C Semantic Web Deployment Working Group and (3) W3C Semantic Web Education and Outreach (SWEO) Interest Group. We foresee that these three groups play an important role in the development of Semantic Web technologies and their best practices.

Furthermore, we also provide some success stories which are related to the best practices. We hope that these success stories could help illustrate some more technical details of the best practices. The connections of between and best practices and success stories presented here are mainly due to researchers in common. This seems to suggest in the future how more success stories could be enabled through transfer of best practices.

In order to spread the messages further and maintain sustainability, we plan to submit the final version of the deliverable and our further work on best practices and related success stories to the Information-Gathering Input Form (http://www.w3.org/2001/sw/sweo/public/Info/) of the W3C Semantic Web Education and Outreach (SWEO) Interest Group. SWEO is currently maintaining some of the wiki pages that SWBPD was maintaining, such as the one on Semantic Web Tools (http://esw.w3.org/topic/SemanticWebTools). Therefore, it makes sense to submit current and future work on best practices and success stories to the SWEO Web site after the Knowledge Web project finishes.

As the Semantic Web standards RDF and OWL are only available since 2004, there will surely be more research and practices about them in the near future. Therefore, our deliverable is simply a first step toward a global document aiming at synthesising success stories and best practices of semantic Web technologies.

In the future, we would like work more on a general framework to encourage the generation of best practices. As we discussed in previous sections, best practices are based on practical experience, consensus or (sometimes) well understood theoretical research. We foresee that study on usability can encourage users to try the technologies, fostering identifying new problems and possible related solutions, so as to encourage further best practices in a long run.

6 References

[Cun02b] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).* Philadelphia, July 2002

[Dim04b] N. Dimitrova, J. Zimmerman, A. Janevski, L. Agnihotri, N. Haas, D. Li, R. Bolle, S. Velipasalar, T. McGee, L. Nikolovska. Media Personalisation and Augmentation Through Multimedia Processing and Information Extraction. In L. Ardissono and A. Kobsa and M. Maybury (eds.*)* *Personalised Digital Television.* Kluwer, pp. 201-233, 2004.

[Maynard05b] D. Maynard, M. Yankova, A. Kourakis and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch, ESWC Workshop "End User Apects of the Semantic Web", Heraklion, Crete, 2005.

[May04b] D. Maynard, M. Yankova, N. Aswani, H. Cunningham. Automatic Creation and Monitoring of Semantic Metadata in a Dynamic Knowledge Portal. *Proceedings of the 11th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA 2004)*, Varna, Bulgaria, 2004.

[NHBL07] L. Nixon, J. Hoppenbrouwers, W. Bille and V. Louis. D1.1.5v3 Communication Channel with Industrial Board and Industry, Knowledge Web Technical Report, 2007.

[PaHo05] Jeff Z. Pan and Ian Horrocks. OWL-Eu: Adding Customised Datatypes into OWL, In Proc. of the Second European Semantic Web Conference (ESWC 2005), pages 153-166, 2005. An extended version appears in the Journal of Web Semantic, 4(1). An online version is available at http://www.websemanticsjournal.org/ps/pub/2005-24.

[PaTh07] Jeff Z. Pan and Edward Thomas. Approximating OWL-DL Ontologies. In Proc. of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07), 1434-1439. 2007.

[Pop02a] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff and M. Goranov, 2004. KIM -- Semantic Annotation Platform, *Journal of Natural Language Engineering*

[Przybocki99] M. Przybocki, J. Fiscus, J. Garofolo and D. Pallett. 1998 {HUB-4} Information Extraction Evaluation, Proceedings of the DARPA Broadcast News Workshop, Herndon, VA, pp.13-18, 1999.

[WaPa05] 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)*, 1279-1295. 2005.

[WaPa06] Shenghui Wang and Jeff Z. Pan. **Integrating and Querying Parallel Leaf Shape Descriptions**. In *Proc. of the 5th International Semantic Web Conference (ISWC2006)*, 668 – 681. 2006.

[WBHL*05] K. Wolstencroft, A. Brass, I. Horrocks, P. Lord, U. Sattler, R. Stevens, and D. Turi. A Little Semantic Web Goes a Long Way in Biology. In *4th International Semantic Web Conference*, volume 3792, pages 786-800, Galway, Ireland, 2005.