

Semantic web technologies as the foundation for the information infrastructure

Frank van Harmelen

Department of Artificial Intelligence, Vrije Universiteit Amsterdam

The Semantic Web is arising over the past few years as a realistic option for a world wide Information Infrastructure, with its promises of semantic interoperability and serendipitous reuse. In this paper we will analyse the essential ingredients of semantic technologies, what makes them suitable as the foundation for the Information Infrastructure, and what the alternatives to semantic technologies would be as foundations for the Information Infrastructure. We will make a survey of the most important achievements on semantic technologies in the past few years, and point to the most important challenges that remain to be solved.

3.1 Historical trend towards increasing demands on interoperability

When Thomas Watson, the founder of IBM, was asked for his estimate of how many computers would be needed worldwide, his reply is widely claimed to have been: ‘about five’

¹. Of course, this presumed reply was given in 1943, but it shows the enormous shift in perspective that has taken place since the very early days of computing. Right until the late 1970's, the dominant perspective on computing was that of mainframe computing: large machines that provided centralised means of computing and data storage. In such a centralised perspective, interoperability is not the main concern: data are locked up in a centralised location, movement of data is rare, and if data is to be integrated, a special purpose ad hoc transformation procedure is applied to transform the data into the required format.

The first revolution that was a major upset to the centralised perspective was the advent of the PC in the 1980's (ironically enough, also dominated by IBM). Suddenly, there were millions of small computing devices, each of which was capable of storing its own data, without recourse to centralised data storage. In this context, interoperability of data was becoming a problem: how to combine the data set stored in (or generated on) one PC with those of another PC, where another user in a different organisation, would be generating their own data?

However, the low degree of connectivity between the different PCs still kept the interoperability problem at bay. It was only the second revolution that really caused the data interoperability problem to bite, namely the advent of the Internet (also arising in the 1980's), culminating in the rapid growth of the Web in the 1990's. The Internet has solved most wide-area networking problems with its nearly universally supported TCP/IP Internet Protocol and its DNS (Domain Name System) host-addressing scheme.

Suddenly, it became possible to exchange information from any computer to any other computer, and between any two users on the planet. In such a setting, special purpose and ad hoc

¹ although this quote is widely questioned now <http://en.wikipedia.org/wiki/Thomas_J._Watson>, it makes the point

transformation procedures to import data are no longer a feasible alternative, and more principled mechanisms to ensure interoperability are required.

3.2 Interoperability at different abstraction layers

The problem of interoperability in any information infrastructure (be it world-wide or local, be it for geographic information or otherwise) can be analysed at different layers of abstraction (also see Chapter 8, Figure 8.1), all of which must be solved in order to obtain full interoperability:

Physical interoperability concerns the lowest layer of the abstraction hierarchy: plug-shapes and sizes, voltages, frequencies, and the bottom layers of the ISO/OSI network hierarchy. This is where most of the progress has been made, and physical interoperability between systems has been solved: with the advent of hardware standards such as Ethernet, and with protocols such as TCP/IP and HTTP (Hypertext Transfer Protocol), we can nowadays walk into somebody house or office, and successfully plug our computer into the network (even automatically via wireless LANs), giving instant world-wide physical connectivity. Ironically, it is the success with which the physical interoperability problem has been solved that now creates problems at higher interoperability levels.

Syntactic interoperability: Physical connectivity is not sufficient. We must also agree on the *syntactic form* of the messages we will exchange. Again, much progress has been made in recent years, particularly with the advent of eXtensible Markup Language isoXML. XML has been dubbed ‘the ASCII of the 21st century’, and indeed it is now the most widely used syntactic standard, and is itself used as a carrier for other syntactic standards such as HTML (for the

content of web-pages¹), WSDL (Web Service Description Language², for the interfaces of web-services), and SOAP (Simple Object Access Protocol³, for the format of web-service messages), *Semantic interoperability*: Of course, even syntactic interoperability is not enough. We need not only agree on the form of the messages we exchange (structure of the information), or the form of the web-pages that we publish, but also need to know the intended meaning of such messages and pages. In case we want also machine processing, e.g. in urgent situations where human decisions have to be supported by machine processing (selections, combinations, translations, and other reasoning tasks), then the intended meaning has to be formalized.

3.3 The meaning of semantic interoperability

In this section we shall be somewhat more precise about the meaning of semantic interoperability. Semantic interoperability is usually defined in terms of a formal semantics, and this can be done either denotational, inferential, or operational. Although the primary definition of the semantics of formal languages is most often in terms of a denotational semantics (e.g. (P. Hayes, 2004) and (Patel-Schneider et al, 2004) for RDF (the Resource Description Framework) and OWL (the Web Ontology Language) respectively, we will instead describe semantic interoperability in terms of inferential semantics.

When an agent (a web-server, a web-service, a database, a human in a dialogue) utters a message, the message will often contain more meaning than only the tokens that are explicitly

¹ <http://www.w3.org/Markup/>

² <http://www.w3.org/TR/wsdl>

³ <http://www.w3.org/TR/soap/>

present in the message itself. Instead, when uttering the message, the agent has in mind a number of ‘unspoken’, implicit consequences of that message. When a web-page contains the message ‘Amsterdam is the capital of The Netherlands’, then one of the unspoken, implicit consequences of this is that Amsterdam is apparently a city (since capitals are cities), that The Hague is not the capital of the Netherlands (since every country only has precisely one capital), etc. If agent A utters the statement about Amsterdam to agent B, they can only be said to be truly semantically interoperating if B not only knows the literal content of the phrase uttered by A, but also understand a multitude of implicit consequences of that statement which are then shared by A and B.

Minimal semantic interoperability: explicit content only

Thus, we could say that the semantic interoperability between A and B increases with the amount of information that they agree on after having exchanged a message. The minimal amount of information that they share is only the fact expressed in the statement itself: there is some object ‘Amsterdam’ and some object ‘The Netherlands’, and they are related by the first ‘being the capital of’ the second. Notice that this minimal amount of semantic interoperability is already non-trivial. Simply exchanging the following arbitrary XML syntax

```
<is-capital-of>
  <Amsterdam/>
  <Netherlands/>
</is-capital-of>
```

is by itself *not* enough for B to understand that we are dealing with two objects and a relation between them: is ‘being capital of’ a relation between two objects (as indeed intended), or does the tree-structure of the XML denote some type-information, as in

```

<humans>
  <males/>
  <females/>
</humans>

```

or does it denote some part-of information, as in

```

<heart>
  <left-chamber/>
  <right-chamber/>
</heart>

```

or any of an infinite number of other plausible semantic interpretations of the same syntactic structure. Thus, even to obtain from the earlier XML

```

<is-capital-of>
  <Amsterdam/>
  <Netherlands/>
</is-capital-of>

```

the minimal intended meaning that we are dealing with a relation between two objects, agents A and B must have previously agreed on this intended meaning of their syntactic structure, namely that the root of the XML tree is the relation between a subject (first subnode) and an object (second subnode). In the context of the Semantic Web, this is exactly the amount of semantic interoperability that RDF enables (that is: RDF without RDF Schema). It allows to pass single sentences, and only the literal content of those sentences themselves are guaranteed to be shared with any other agent adhering to the RDF semantics. Of course, the precise syntactic encoding is arbitrary (as long as it is agreed upon), and we could have written the above as

```

<relation name='capital-of'>

```

```

    <object name='Amsterdam' />
    <object name='The Netherlands' />
  </relation>

```

or indeed as

```

<rdf:Description rdf:about='3116'>
  <name>Amsterdam</name>
  <isCapitalOf>The Netherlands</isCapitalOf>
</rdf:Description>

```

as it would read in RDF syntax.

Extended semantic interoperability: shared inferences

So although obtaining even minimal semantic interoperability from purely syntactic structures is non-trivial, it is of course a very limited form of semantics. In any reasonable human conversation, saying that 'Amsterdam is the capital of The Netherlands' would also imply a number of other, unspoken facts, implicitly implied by what was said: that Amsterdam is apparently a city (since capitals are cities), that The Hague is not the capital of the Netherlands (since every country only has precisely one capital), that The Netherlands is a country, or a province, but not another city, since countries and provinces have capitals, but cities don't. A spatial implied fact is the location of the capital city is inside the area covered by the country. Thus, a more extended form of semantic interoperability would guarantee that if agent A utters a sentence S to B, then not only does B believe the literal contents of S, but B should also believe a number of other facts that can be inferred from S in combination with shared knowledge between A and B. It is exactly this shared knowledge that has become known as the shared *ontology*

between A and B. In our little example, if such an ontology would indeed capture the fact that capitals are cities, capitals are unique, countries have capitals, etc, then A and B are guaranteed to have much better basis for exchanging the intended meaning ('semantics') of sentence S beyond its limited literal content. In fact, we could say that the amount of semantic interoperability between A and B is measured by the number of new facts that they both subscribe to after having exchanged a given sentence: the larger and richer their shared ontology, the more semantically interoperable they are.

It is exactly this kind of shared ontological information that can be captured in RDF Schema (as opposed to RDF only):

```
<rdfs:Class rdf:about='Capital'>
  <rdfs:subClassOf rdf:resource='#City' />1
</rdfs:Class>

<rdf:Property rdf:ID='isCapitalOf'>
  <rdfs:domain rdf:resource='#Capital' />
  <rdfs:range rdf:resource='#Country' />
</rdf:Property>
```

states that capitals are cities, and that capitals are capitals of countries (allowing to infer that Amsterdam must be a city if it is the capital of The Netherlands).

A more expressive language such as OWL is required to express that the capital of a country is

¹ where #City is a shorthand URI referring to the concept City, defined at the same location as where the above statement can be found.

unique:

```
<owl:InverseFunctionalProperty rdf:ID='isCapitalOf' />
```

(the semantics of `InverseFunctionalProperty` states that the value of such a property uniquely defines the object of the property, since the inverse property (from value to object) is functional (has exactly one value).

Full semantic interoperability: upper and lower bounds

OWL is more expressive than RDF Schema in a very specific way: when agreeing on an ontology *O* expressed in RDF Schema, two agents *A* and *B* have both committed to a *minimal* set of beliefs that they will both uphold given some sentences *S* to be exchanged in the future; again, in our example, if *A* states that Amsterdam is the capital of The Netherlands, then by subscribing to their shared ontology *O*, *B* is forced to believe a number of other things as well (Amsterdam being a city, etc). Hence, there is a minimum set of beliefs, a *lower bound*, on what agents *A* and *B* will infer after having exchanged a sentence *S*.

However, using an RDF Schema ontology, *A* cannot *forbid* *B* to believe certain things, for example it cannot forbid *B* to believe that besides Amsterdam, The Hague is also a Dutch capital. Technically, this amounts to saying that RDF Schema cannot express negative information, it does not contain negation. OWL does, hence in OWL, we can say that if Amsterdam is the Dutch capital, no other city can be (the `InverseFunctionalProperty` above). OWL enables *A* and *B* to not only put a lowerbound on what they must believe after exchanging a sentence, it also allows them to put an *upperbound* on what they may not believe after exchanging a sentence. By strengthening the ontology *O*, *A* and *B* can move these lowerbound and

upperbound successively closer together, hence narrowing the window of opportunity for any misunderstandings (consisting of things that one of them believes after S, while the other one does not). Stronger ontologies, that place higher lowerbounds and lower upperbounds on the set of inferred consequences of an exchanged sentence, increase the semantic interoperability between two agents.

The role of XML as a notation

The above semantic descriptions have all been given in the XML notation that is prescribed by the W3C standards ((Bechhofer et al., 2004) for OWL and (Becket, 2004) for RDF). However, this is only *one* particular syntax in which we can state the intended semantics. Of course, the XML syntax has a special status, since it is the one that was chosen for the standardisation documents. Nevertheless, different syntactic forms exist for expressing the same semantic contents. Examples in the case of RDF and OWL are the N3 syntax¹ that is popular because it is much more compact and readable than the official XML syntax. Also very popular is the UML-based notation because of its graphical presentation. In fact, many of the chapters in this volume (e.g. chapters 3,4,5,9) use UML class-diagrams to present knowledge that is also easily formalisable in OWL's official XML syntax.

3.4 The Semantic Web as a foundation for an Information Infrastructure

We are now in a position to define the role of Semantic Web technology as a foundation for an Information Infrastructure. The Semantic Web offers technology that contributes towards solving

¹ <http://www.w3.org/2000/10/swap/Primer>

the interoperability problem at all three of the layers discussed above: HTTP, DNS and URI's for physical interoperability, XML for syntactic interoperability, and RDF, RDF Schema and OWL for semantic interoperability.

These languages and their corresponding technology are organised in a stack, where each higher layer uses lower ones. Semantic Web applications achieve semantic interoperability by exchanging not only their data, but by also exchanging (or having previously agreed to) explicit models of these data. These shared data models are often known as ontologies, and constitute shared knowledge used to interpret the information to be exchanged.

The most important premise on which the Semantic Web rests can be now be phrased as follows:

Premise: In order to achieve semantic interoperability it is certainly necessary (and most likely also sufficient) to express both data and data-models (a.k.a. ontologies) in languages with a formalised semantics, which enforce the sets of beliefs that agents must or may not uphold as the result of exchanging a certain piece of information.

Notice that this premise is very close to the Knowledge Representation Hypothesis, formulated by Brian Smith in his 1982 Ph.D. thesis (Smith, 1982):

‘Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge.’

We can recognise ‘the propositional account of knowledge’ in the propositional structure of the Semantic Web languages (RDF, RDF Schema, OWL), and the ‘causal and essential role in engendering behavior’ is similar to the inference-enforcing process based on shared background

knowledge that we discussed in the previous section.

Alternative approaches

We should note that although this approach to semantic interoperability may sound plausible, it is certainly not the only possible route. In particular the possibility of a propositional account of the required knowledge to express sufficiently rich data-models has traditionally been criticised in Knowledge Representation, and similarly in the Semantic Web. And it must be acknowledged that the currently most effective approaches to search are not based on *propositional* accounts of the contents of the Web, but rather on *statistical* models, as used in search engines such as Google, using word frequencies, and patterns of links between pages instead.

Although the two approaches (the propositional and the statistical) are often positioned as alternatives, there is in fact nothing to make them mutually exclusive. It is well possible for example to imagine statistical patterns being used as the basis for constructing an propositional account, and in fact many machine-learning contributions to Semantic Web technology (e.g. ontology learning) take exactly this combined approach.

3.5 Most important achievements to date

After having outlined the foundational ideas underlying the Semantic Web, and having described their role in a semantically interoperable Information Infrastructure, in this section we will discuss the most important achievements in recent years towards the realisation of this Semantic Web Information Architecture.

Ontology languages

A crucial and widely known achievement has been the definition and adoption of a number of data-modelling languages, stacked one on top of the other, with ever more expressivity: RDF, RDF Schema, OWL, where the latter is itself divided into three substrata with proper syntax and semantic inclusions. Without going into the details of these language (ample reference and teaching material exists), the general power of these languages is as follows:

- *RDF*: expressing binary relations between objects, and expressing that an object belongs to a given type (or ‘class’);
- *RDF Schema*: arranging these classes and properties in a class and property inheritance hierarchy (superclass-subclass), and stating that properties have certain types as their domain and range;
- *OWL Lite*: expressing (in)equalities between individuals, between classes and between properties, and stating algebraic properties of properties (transitivity, symmetry, inverse functionality, etc), 0/1-restrictions on the cardinality and ranges of properties;
- *OWL DL*: definition of classes by enumeration, algebraic operations on classes (intersection, union, complement), stating disjointness of classes, arbitrary cardinality restrictions on properties
- *OWL Full* introduces no new language constructions, but is more liberal in the way these constructions are combined (for example using classes as instances of other classes).

Of course the increase in expressivity comes with an increase in computational costs of doing

inference in these languages. The above stack of languages allows users to pick the language with the appropriate cost/benefit trade-off for each particular application. The (almost) proper inclusion relations between these languages ensures the possibility to move to more expressive languages as and when the need arises, without having to redo prior efforts.

With their status of W3C recommendation, these languages are guaranteed to be implementable and stable, enabling a rapid growth of industrial investment in their support and deployment.

Ontology vocabularies

The above Semantic Web data-modelling languages have indeed been used for the construction of data-models in a wide variety of domains. Very often, this was not a construction from scratch, but instead involved translating previously existing structures ('datamodels', 'ontologies', 'thesauri', 'vocabularies') into these new languages with their standardised syntax and semantics, thereby illustrating how these languages do indeed enable greater degrees of semantic interoperability.

It is already impossible to be exhaustive (the Swoogle Semantic Web search engine lists more than 10.000 different vocabularies at the time of writing). We give the following incomplete list only to illustrate the diversity of ontologies expressed in Semantic Web languages, both in choice of domain and in how extensive the modelling has been

- bio-medical: GO (15.000 terms from molecular biology¹), SNOMED (300.000 terms

¹ <http://www.geneontology.org/>

from general medicine¹), UMLS (a loosely integrated collection of over 100 medical vocabularies²), FMA (205,000 concepts describing anatomy³)

- top-level: Cyc (hundreds of thousands of terms and millions of assertions capturing common sense knowledge⁴), SUMO (20.000 terms and 60.000 axioms capturing common sense knowledge plus various specialised domain⁵), WordNet (115.000 definitions for about 150.000 words from the English language⁶)
- Cultural Heritage: AAT (34,000 concepts, and 131,000 terms relating to fine art, architecture, decorative arts, archival materials, and material culture⁷), IconClass (28,000 definitions of objects, persons, events and abstract ideas that can be the subject of an image⁸), ULAN (293,000 names and biographical and bibliographic information about artists and architects⁹)
- Geographical: See chapter 3 of this volume by Lieberman and Goad for a description of a number of geographical vocabularies and ontologies.

Of course, these languages and vocabularies are only useful if they are used *for* something. In the next section, we will describe tools for using these languages and vocabularies, followed by

¹ <http://www.snomed.org/>

² <http://umlsinfo.nlm.nih.gov/>

³ <http://fma.biostr.washington.edu/>

⁴ <http://www.cyc.com/>

⁵ <http://www.ontologyportal.org/>

⁶ <http://wordnet.princeton.edu/>

⁷ http://www.getty.edu/research/conducting_research/vocabularies/aat/

⁸ <http://www.iconclass.nl/>

⁹ http://www.getty.edu/research/conducting_research/vocabularies/ulan/

some brief examples of successful use-cases that were built using these languages, tools and vocabularies.

Tools

The above vocabularies, and others which are routinely being built these days in many different application areas, are far too large and complicated to be managed manually. The community has developed a large set of methods and tools for creating, managing and deploying such large ontologies. Because of the rapidly evolving state of the art there is little point in putting any list of such tools in print, but they cover every aspect of an ontology's life cycle:

- *creation* (either through knowledge acquisition and manual modelling, or through concept extraction from a corpus of text, or through machine learning from a large dataset),
- *change management* (detecting which changes have occurred between versions, alerting for possible inconsistencies or redundancies this may have caused)
- *modularisation* (= selecting the right subvocabulary for a given task),
- *ontology alignment* (= integration of multiple vocabularies for a single use),
- *storage and querying* of very large datasets organised by an ontology (at the time of writing, ontology stores can handle in the order of billions of facts),
- *reasoning* (= drawing inferences from the given facts using the reasoning steps that are allowed under the formal semantics of the language)
- *visualisation* (= visualising large datasets organised by ontologies, either in the form of tree-diagrams, in the form of other diagrams such as cluster-maps, see Figure 3.1).

Many of these tools have outgrown the stage of academic prototyping, and are available as commercial software, including support services.

Annotation and classification techniques

Of course, such large ontologies are only useful if used to describe and organise large datasets (the datasets between which we want to obtain semantic interoperability in the first place). This involves a task called *annotation* or *classification* in different parts of the community, but these are essentially the same task: given an item from a dataset and given an ontology for the domain of the dataset, decide which class(es) the data item must be assigned to. This task is of course crucial to the use of ontologies for solving semantic interoperability problems: if a data-item is not assigned to a class in the ontology, it is not known to the receiving agent what to make of this data-item, and which inferences to draw about this item.

This crucial task can be done either manually or automatically (in restricted domains).

- *manual*: in a large number of domains, manual annotation will remain the dominant mode of classification for some time to come. In particular for non-text items (sound, still images, video), automatic classification remains very hard (e.g. Snoek et al, 2007). In many audio-visual archives (e.g. in the cultural sector, satellite images, aerial photography, but also in medical applications), software uses an ontology to suggest annotations to users, but the final annotation is only made after user approval. The same holds for high quality long-term archives such as National Libraries (van Gendt et al, 2006).

- *automatic*: if an ontology is expressed in OWL DL, it is possible to determine the necessary and (sometimes) the sufficient properties that a data-item must satisfy to belong to a certain class in the ontology. For text corpora, or for semi-structured datasets, it is possible to automatically analyse the properties of data-items and then determine to which class(es) they belong.

The tasks of annotating, classifying and performing inference are often integrated into a single environment. Tools such as Sesame¹ or the Oracle² tools set give a single integrated environment in which to write down semantic definitions of classes and their relations, store instances of those classes, and perform reasoning with such instances and classes.

Use-cases, scenarios, applications

All the above languages, tools and technology have been used to develop showcases in a variety of domains. Without any claim to completeness, we will sketch here a small number of use-cases which illustrate in particular the role of semantic web technology in semantic interoperability.

Together, these use-cases do indeed make it credible that Semantic Web technology can indeed serve as the foundation for semantic interoperability in a world-wide Information Infrastructure.

eCulture browser over multiple art collections: In the eCulture browser¹ a number of art collections from major Dutch musea are manually annotated using a number of different ontologies (VRA, and the Getty thesauri AAT, ULAN and TGN). Manual links that were

¹ <http://www.openrdf.org>

² http://www.oracle.com/technology/tech/semantic_technologies/index.html

established between these different ontologies then allowed a single faceted-browsing interface across the different collections. A similar result (although on a much smaller scale) was obtained in the STITCH browser² where illuminations from medieval manuscripts in the Dutch and French National Libraries are shown in a single interface, using automatically created links between the thesauri that were used to index the separate collections. An additional feature of the STITCH browser is that because one of the thesauri is multi-lingual, it becomes possible to search for manuscripts using search terms in a different language than with which they were annotated, using the multi-lingual thesaurus as a translation device. This is also very relevant in the spatial context; e.g. in INSPIRE there should be support for 21 languages; see chapter 1.

DOPE browser over multiple scientific literature collections: In the DOPE project (the Drug Ontology Project for Elsevier, Stuckenschmidt et al, 2004), a single large thesaurus (EMTREE, a commercial product by Elsevier, 45.000 preferred terms and 190.000 synonyms to describe mainly drugs and diseases) was used to index a large body of scientific literature (5 million abstracts from the Medline database and another 500.000 full text papers from Elsevier's Science Direct collection). Annotation of papers and abstracts with EMTREE terms was done automatically using commercially available concept extract techniques. These semantically indexed heterogeneous collections could then be browsed using a single interface (see Figure 3.1), which used the ontology for query disambiguation, hierarchical and clustered display of search results, and query refinement.

¹ <http://e-culture.multimedien.nl/demo/search>

² <http://stitch.cs.vu.nl>

The screenshot shows the DOPE Browser interface. On the left, the 'Focus Term' section displays 'acquired immune deficiency ...' and a search for 'aids'. Below this is a 'Co-occurring Terms' list with a tree structure. The 'chemicals and drugs' category is expanded, showing terms like 'CD4 antigen' (55), 'zidovudine' (37), and 'idoxuridine' (13). The 'Term Overlap Display' in the center shows a network of colored nodes: a red node for 'zidovudine (37)', a green node for 'idoxuridine (13)', and a large blue node for 'CD4 antigen (39/55)'. The 'Document List' at the bottom shows 'Contents of the cluster of "CD4 antigen":' with two entries, including a reference to a 1999 article in *Nutrition* 15 (6), pp. 453-457, with a URL: <http://linkinghub.elsevier.com/pii/S0899900799000830>.

Figure 3.1 Interface of the DOPE browser, showing ontology-based hierarchical and clustered display of search results using a single ontology across multiple collections (the color coding of individual objects reflects different types of publications: journal, conference, survey, etc).

The *Semantic Web Education and Outreach (SWEO) Interest Group* from W3C has published a collection of a few dozen use cases of semantic technologies¹, ranging from eGovernment to eCommerce, and from improved search to data-integration, covering sectors as diverse as the automotive industry, the financial sector, the IT industry, the publishing world and others.

¹ <http://www.w3.org/2001/sw/sweo/public/UseCases/>

Geosemantic applications

One of the use-cases in the SWEO collection describes the use of semantic technologies by the British Ordnance Survey¹ to cut the cost and improve the accuracy of data integration. By using an ontology of Ordnance Survey's data, the semantic differences between different data-sets are made explicit, thus facilitating data-integration, both among OS datasets and between data-sets of OS and its customers. An ontology has been built for the Hydrology domain, as well as an Administrative Geography for Great Britain in RDF. An ontology for an ontology for Buildings and Places is in progress². Similar work on semantic translations of cadastral information has been reported in (Hess & de Vries, 2006). Closely related work, but aimed at the integration and chaining of geographic *services* as opposed to geographic datasets is reported in (Lemmens et al, 2006). Other work aiming at the integration of geographic services has earned a top ranking in the 2006 Semantic Web Challenge³.

3.6 The most important challenge: ontology mapping

Perhaps the most important challenge of all is how to deal with semantic interoperability across multiple ontologies: although a shared ontology gives a way to draw shared inferences that explicate the intended meaning of data-items, this does require a *shared* ontology. Semantic interoperability across multiple ontologies requires to align these different ontologies. This problem has been the subject of research in different fields over many decades. Different variants

¹ <http://www.w3.org/2001/sw/sweo/public/UseCases/OrdSurvey/>

² All of these available at <http://www.ordnancesurvey.co.uk/ontology>

³ <http://www.laits.gmu.edu/geo/nga/>

of the problem received names such as "record linkage" (dating back to Newcombe's work on linking patient records (Newcombe, 1959), and surveyed in (Winkler, 1999), schema integration (Rahm, 2001), and more recently ontology mapping (see the recent book (Euzenat & Shvaiko, 2007) for what is currently the best survey of the state of the art).

An important development in this historical progression is the move towards ever richer structure: the original record linkage problem was defined on simple strings that were names of record-fields; the schema-integration problem already had the full relational model as input; while ontology mapping problems are defined on full hierarchical models plus rich axiomatisations. Each step in this progress has all the solutions of the previous steps at its disposal (since each later model subsumes the earlier ones), plus new methods that can exploit the richer structures of the objects to be aligned.

Current approaches to ontology mapping deploy a whole host of different methods, coming from very different areas. These can be categorised to distinguish linguistic, statistical, structural and logical methods. The currently available best survey of ontology alignment techniques is (Euzenat & Shvaiko, 2007).

Linguistic methods are directly rooted in the original record linkage work all the way back to the early 60's. They try to exploit the linguistic labels attached to the concepts in source and target ontology in order to discover potential matches. This can be as simple as basic stemming techniques or calculating Hamming distances, or can use specialised domain knowledge.

Statistical methods typically use *instance data* to determine correspondences between concepts: if there is a significant statistical correlation between the instances of a source-concept and a target-concept, there is reason to believe that these concepts are strongly related (by either a

subsumption relation, or perhaps even an equivalence relation). These approaches of course rely on the availability of a sufficiently large corpus of instances that are classified in both the source and the target ontology.

Structural methods exploit the graph-structure of the source and target ontologies, and try to determine similarities between these structures, often in coordination with some of the other methods: if a source- and target-concept have similar linguistic labels, then dissimilarity of their graph-neighbourhoods can be used to detect homonym problems where purely linguistic methods would falsely declare a potential mapping.

Logical methods are perhaps most specific to mapping *ontologies* (instead of mapping record-fields or database-schemata). After all, in the time-honoured phrase of (Gruber, 1993), ontologies are "*formal specifications of a shared conceptualisation*" (my emphasis), and it makes sense to exploit this formalisation of both source and target structures. A particularly interesting approach is to use a third ontology as background knowledge when mapping between a source and a target ontology: if relations can be established between source (resp. target) ontology and different parts of the background knowledge, then this induces a relation between source and target ontologies. A serious limitation to this approach is that many practical ontologies are rather at the semantically lightweight end of Uschold's spectrum (Uschold, 1996), and thus don't carry much logical formalism with them.

Given the difficulty of the problem, and the amount of work already spent on it, it seems unlikely that the problem of ontology mapping will yield to a single solution. Instead, this seems more the kind of problem where many different partial solutions are needed. Currently, our toolbox of such partial solutions is already quite well stocked, and is still rapidly growing. However, a theory of which combination of partial solutions to apply in which circumstances is still lacking.

3.7 Conclusion

Undoubtedly, the problem of semantic integration is one of the key problems facing Computer Science today. Despite many years of work, this old problem is still open, and has actually acquired a new urgency now that other integration barriers (physical, syntactic) have been largely removed.

The ontology-based approach of the Semantic Web with its richer datamodels (logic-based hierarchical ontologies instead of the flat relational models), which allow rich inferences to be made, is a promising foundation for semantic interoperability in the Information Infrastructure.

Bibliography

Cees G.M. Snoek, Bouke Huurnink, Laura Hollink, Maarten de Rijke, Guus Schreiber, Marcel

Worring. Adding Semantics to Detectors for Video Retrieval. *IEEE Transactions on Multimedia*, 9(5):975-986, August 2007.

Hayes, P., RDF Semantics, W3C Recommendation 10 February 2004,

<http://www.w3.org/TR/rdf-mt/>

Patel-Schneider, P., P. Hayes, I. Horrocks, OWL Web Ontology Language, Semantics and

Abstract Syntax, W3C Recommendation 10 February 2004, [http://www.w3.org/TR/owl-
semantics/](http://www.w3.org/TR/owl-
semantics/)

van Gendt, M., A. Isaac, L. van der Meij and S. Schlobach. 2006, Semantic Web Techniques for

Multiple Views on Heterogeneous Collections: a Case Study. Proceedings of the *10th European Conference on Research and Advanced Technology for Digital Libraries*

- (ECDL 2006), Julio Gonzalo, Constantino Thanos, M. Felisa Verdejo and Rafael C. Carrasco (eds.), Springer Verlag, LNCS vol. 4172, pp. 426-437. Alicante, Spain, September 17-22, 2006
- Stuckenschmidt, H. , F. van Harmelen, A. de Waard, T. Scerri, R. Bhogal, J. van Buel, I. Crowlesmith, Ch. Fluit, A. Kampman, J. Broekstra and E. van Mulligen, 2004, 'Exploring Large Document Repositories with RDF Technology: The DOPE Project', *IEEE Intelligent Systems*, 2004, Vol. 19, No. 3, pgs. 34-40.
- Newcombe, H.B., J.M. Kennedy, S.J Axford, and A.P. James, 1959, Automatic linkage of vital records. *Science*, 130:954–959, 1959.
- Winkler, W., 1999, The state of record linkage and current research problems. *Technical report*, Statistical Research Division, U.S. Bureau of the Census, Washington, DC, 1999.
- Rahm, E. and P. A. Bernstein. 2001, A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, 10(4):334–350, 2001.
- Euzenat, J. and P. Shvaiko, 2007, *Ontology Matching*, Springer Verlag, 2007
- Gruber, T., 1993, A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–200, 1993.
- Uschold, M. and M. Gruninger, 1996, Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
- Bechhofer, S. F van Harmelen, J Jim Hendler, I Horrocks, D. McGuinness, P. Patel-Schneider, S. Stein, OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004, <http://www.w3.org/TR/owl-ref/>
- Becket, D., 2004 RDF/XML Syntax Specification (Revised), W3C Recommendation 10

February 2004, <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>

Smith, B., 1982, Reflection and Semantics in a Procedural Language, *PhD thesis*, M.I.T, 1982,
Tech. Report MIT-LCS-TR-272.

Hess, C. and M. de Vries, 2006, From models to data: A prototype Query Translator for the cadastral domain, In: *Computers, Environment and Urban Systems*, Volume 30 (2006), pp. 529-542

Lemmens, R., A. Wytzisk, R. de By, C. Granell, M. Gould and P. van Oosterom, 2006,
Integrating Semantic and Syntactic Descriptions to Chain Geographic Services. In: *IEEE Internet Computing*, Volume 10, 5, pp. 42-52.