The Roles of Natural Language and XML in the Semantic Web

Graham Wilcock, Paul Buitelaar, Antonio Pareja-Lora, Barrett Bryant, Jimmy Lin and Nancy Ide

1 Introduction *Graham Wilcock*

This chapter is based on the panel session "The Roles of Natural Language and XML in the Semantic Web" at the 2nd Workshop on NLP and XML (NLPXML-2002), held at COLING-2002 in Taipei. The workshop[1] covered a wide range of topics in which both NLP and XML are central: XML-based NLP tools, corpus annotation standards, XML in document generation, XML in spoken dialogue systems. Whereas most of the workshop papers presented tools and systems that are already implemented, the main aim of the panel session was to look ahead to the future development of the Semantic Web. The panel members, nevertheless, were researchers who already have practical experience of using Semantic Web technologies.

The chapter contains six sections, each written independently. I chaired the panel session and wrote this introduction. The next four sections were contributed by the four panel members: Paul Buitelaar presents an overall vision of the Semantic Web and its implementation technologies; Antonio Pareja-Lora describes experiences with XML and RDF; Barrett Bryant describes experiences with DAML+OIL; Jimmy Lin argues for a wider role for natural language. The final section by Nancy Ide forms both a conclusion and a link to the 3rd workshop[2] in the NLPXML series.

In this introduction I outline why we had a panel on "The Roles of Natural Language and XML in the Semantic Web". The Semantic Web has become an important topic, but what do we mean by "the role of XML" and "the role of natural language"? What kinds of roles are required in the Semantic Web? The sections by the panel members will give more detailed descriptions of some of these roles.

One of the questions raised at the panel session, leading to extensive discussion, concerned the possibility or impossibility of having a single all-embracing ontology as the basis for the Semantic Web. I will summarize some of the points from this discussion.

1.1 The Roles of XML

XML[3] (eXtensible Markup Language) can be used in two distinct ways, which are both important for the Semantic Web. First, XML is a language which can be used, directly and by itself, to represent information. Second, XML can be used to define more specialized languages. In fact, XML has been used as the basis of many different specialized languages. Here are just a few examples:

- MathML[4] (Mathematics Markup Language) for mathematical formulae
- WML[5] (WAP Markup Language) for WAP mobile phones

[1] http://www.ling.helsinki.fi/~gwilcock/NLPXML
[2] http://www.cs.vassar.edu/~ide/events/NLPXML3
[3] http://www.w3.org/XML/
[4] http://www.w3.org/Math/
[5] http://www.oasis-open.org/cover/wap-wml.html

- JSML[6] (Java Speech Markup Language) for speech synthesizers
- SVG[7] (Scalable Vector Graphics), a language for XML-based graphics
- XHTML[8], a form of HTML that conforms to XML syntax.

In addition, XML Schema[9] is a language for defining the permitted structures and data types of an XML document type. The schema definition, itself an XML document, is used for validating the contents of other XML documents. An XML Schema can therefore be used to define a specialized language like XHTML. Unlike the earlier DTD form of document type definition, an XML Schema itself conforms to XML syntax. A language that conforms to XML syntax (like XHTML but unlike ordinary HTML) can be efficiently parsed, validated and transformed by standard XML processors.

Among this multitude of XML-based languages there are only a few languages (XML itself, XML Schema, RDF and RDF Schema, DAML+OIL, OWL) that we are concerned with here because they are used particularly in the Semantic Web. In fact, when we talk about "the role of XML in the Semantic Web" we are really using "XML" to refer to this small group of XML-based languages which play a particular role in the Semantic Web. This group of Semantic Web languages and their relationships are described by Paul Buitelaar in Section 2.

1.2 The Roles of Natural Language

Natural language is, of course, ordinary language like English or Chinese. We use the term "natural language" in order to explicitly exclude artificial languages like Java or XML. When we talk about "the role of Natural Language in the Semantic Web" we are referring to the use of natural languages to play some particular role in the Semantic Web, as opposed to the use of the group of XML-based languages (XML, RDF, DAML+OIL) mentioned above.

Of course, the existing World Wide Web already contains enormous amounts of natural language in the texts of many millions of web pages. The problem is that it is difficult to find relevant information and extract it from this huge mass of texts. Most of the texts are marked up in HTML, but the markup mainly specifies the presentation format of the text, not its contents. By contrast, the vision of the Semantic Web is to mark up the semantic content of the information on the Web. The information whose semantic content needs to be marked up may be in many different forms. In addition to natural language texts, the information may be in table format, or in graphical images, audio, video or other forms.

This leads to an interesting question. What form should the markup itself take? Should the markup language for the Semantic Web be XML? Or should it be one of the XML-based languages such as RDF or DAML+OIL, or some combination of these? Or would

[6] http://java.sun.com/products/java-media/speech/forDevelopers/JSML/
[7] http://www.w3.org/TR/SVG/
[8] http://www.w3.org/TR/xhtml1/
[9] http://www.w3.org/XML/Schema

it be better to use natural language as the markup language? When we talk about "the Roles of XML and Natural Language in the Semantic Web" we are referring to this question about what form the markup language should take, not merely to the existence of natural language texts in the Web.

1.3 Layers in the Semantic Web

Sometimes the Semantic Web is described in terms of a layer model. There are different versions of this model, such as the one by Tim Berners-Lee in Figure 1.

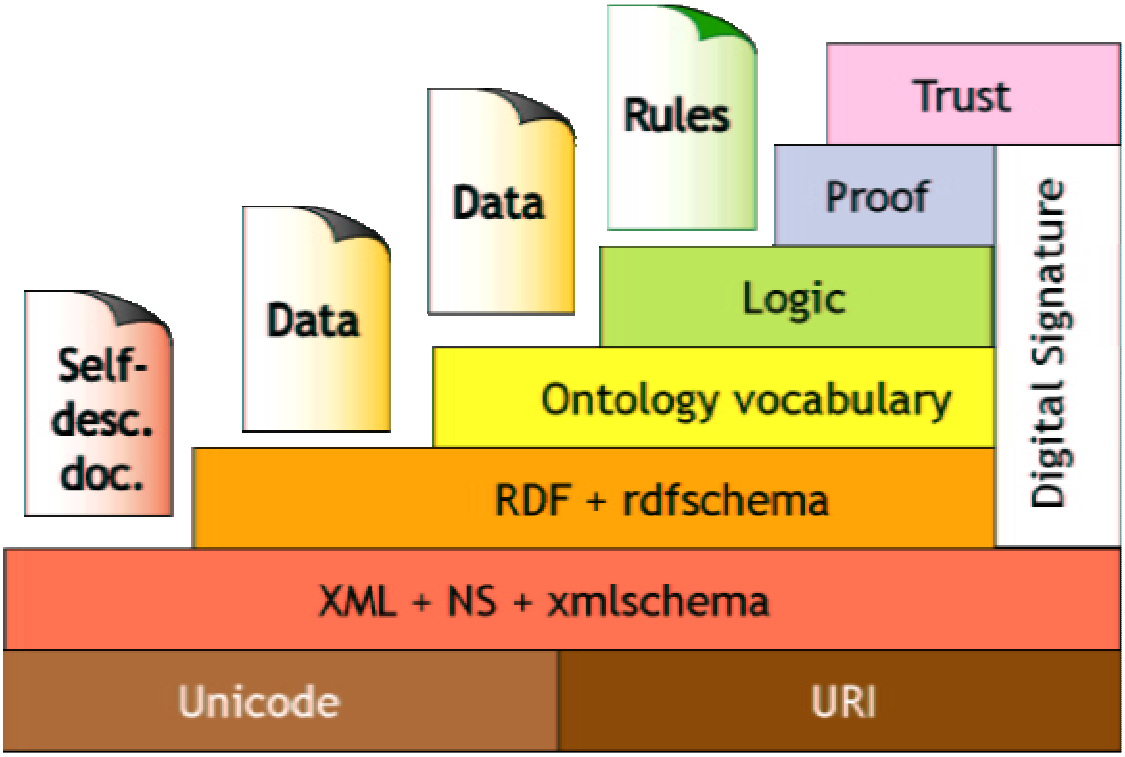


Figure 1-1: Semantic Web Layers[10]

The layers we are most interested in are the XML layer, the RDF layer, and the ontology layer. We are not concerned here with the underlying Unicode/URI layer, and we do not have much to say about the upper layers dealing with logic, proof and trust.

To some extent, the different sections in this chapter focus on different layers. Paul Buitelaar introduces the overall architecture and describes the relationship between the different representation languages. Antonio Pareja-Lora deals mainly with the XML and RDF layers, while Barrett Bryant discusses DAML+OIL used in the ontology layer. As an alternative to these XML-based languages, Jimmy Lin puts forward an alternative proposal for a greater use of natural language.

[10] http://www.w3c.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

1.4 Ontologies

The question at the panel session which generated the most discussion, for which we are indebted to Eduard Hovy, concerned the possibility or impossibility of having a single all-embracing ontology as the basis for the Semantic Web. Where does this question come from? I believe it arises partly from the overall vision of the Semantic Web, and partly from the arguments put forward for the benefits which the Semantic Web will bring. The Semantic Web is based on the World-Wide Web, which is a single all-embracing web, so the vision of the Semantic Web is that it will be a single all-embracing Semantic Web. Among the arguments put forward for the expected benefits of the Semantic Web is the vision that, once all the information in it is semantically annotated clearly and unambiguously in some agreed way, the Semantic Web will make smart searching and inferencing possible, and will eliminate all the confusion and contradictions in the present chaotic World-Wide Web. This annotation will be based on ontological classification, and it will only be fully effective once the Semantic Web reaches a certain "critical mass".

This argument can clearly be challenged from a practical point of view, as the problems involved in building such a semantically annotated web on a global scale are enormous. However, the question raised at the panel session was not whether a single all-embracing ontology is difficult in practice, but whether it is even theoretically feasible. As an example, Eduard Hovy mentioned the question: how many colours are there? It is well-known that different languages and cultures divide the spectrum in various different ways, and there is no possible way to arrive at one single globally agreed list of colours. Similarly, different languages and cultures divide everything else in the world in different ways, and there is no possibility of a global, fundamental agreement about classification of time and space, or entities and events – in short, it seems that there is no possibility of a single all-embracing ontology.

One way to approach this issue is to remember the origins of the World-Wide Web and the Internet. The World-Wide Web is a single all-embracing web, based on the Internet which is a single all-embracing network, but the Internet grew from many small local networks based on local communications protocols, which were gradually joined together into regional and national networks and in some cases into industry-specific or sector-specific networks by adopting agreed protocol standards. Eventually these large networks joined together globally by internetworking based on the Internet Protocol. Similarly, the Semantic Web is growing from many applications based initially on small ontologies stored in local databases. These local ontologies will be combined with others to produce national or industry-specific or sector-specific ontologies, by adopting ontology standards and making the agreed ontologies more widely available. These wider ontologies will become part of the Semantic Web, which will have a sufficient critical mass to produce the benefits predicted. This growth process will be accelerated because the vital importance of standards has been understood, and local applications based on local ontologies are already using W3C standard ontology languages from the outset. The ideas and experiences of some who have pioneered the use of these technologies are presented in the following sections.

2 The Semantic Web: Vision and Implementation *Paul Buitelaar*

2.1 The Semantic Web Vision

The Semantic Web is a vision of a future version of the World-Wide Web, in which all web-based knowledge is encoded in an explicit, formal way to allow for increasingly intelligent and therefore autonomous agents (Berners-Lee et al. 2001).

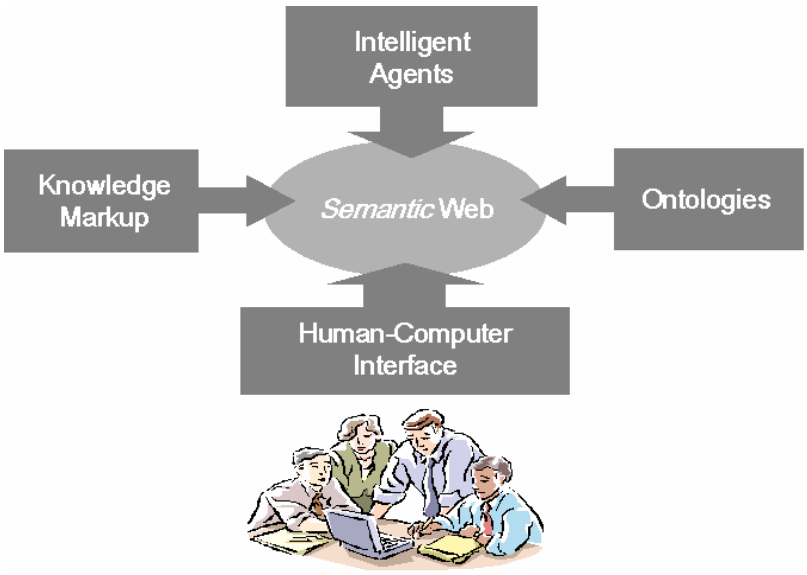


Figure 2-1: The Semantic Web Vision

As illustrated in Figure 2-1, this entails the definition of formal, web-based *ontologies* to express the knowledge that is understood by humans as well as agents, and *knowledge markup* of (textual, multimedia) documents and databases using these ontologies. Knowledge markup is an elaboration of so-called *metadata* as currently defined and in use for a restricted set of applications, e.g. the Dublin Core[11] set of bibliographical metadata such as 'title, 'author', etc. It is to be expected that over the next decade the knowledge structures of many more such applications will be formally encoded in web-based ontologies. Specifically in the context of e-business this will become apparent, as companies will need a common and explicit understanding of their products and services in order to allow for an automatic commercial exchange by artificial agents.

[11] http://dublincore.org/

2.2 The Implementation of the Semantic Web

The Semantic Web is not a new technology in itself, but rather a vision of how existing technologies could be combined in establishing a more intelligent interaction with web-based information. These technologies consist of 'core' technologies in knowledge markup (i.e. markup languages, knowledge representation) and knowledge processing (i.e. intelligent agents, web services) and 'enabling' technologies in knowledge organization (i.e. information science, machine learning) and knowledge access (i.e. database systems, language technology). In the next sections we will discuss the relationship between these technologies and the semantic web in some more detail.

2.2.1 Knowledge Markup and Processing

The definition of web-based knowledge representation languages is currently an active field of study, which has led to a number of proposals and emerging standards. Foremost among these are RDF Schema[12] and DAML+OIL[13] (recently redefined as OWL[14]), the latter of which is defined on top of the other. Besides these, also XML Schema[15] and Topic Maps[16] are sometimes seen as knowledge representation languages.

In Figure 2-2, an overview is given of some important aspects of the XML/RDF family of knowledge markup languages -- overview based on (Gil and Ratnaker 2001). From a syntactic point of view, RDF is written in XML, whereas DAML+OIL is written in RDF. On the semantic side, ontologies written in XML Schema, RDF Schema or DAML+OIL are all based on the notion of a namespace, which defines the interpretation context of any XML, RDF or DAML+OIL expression.

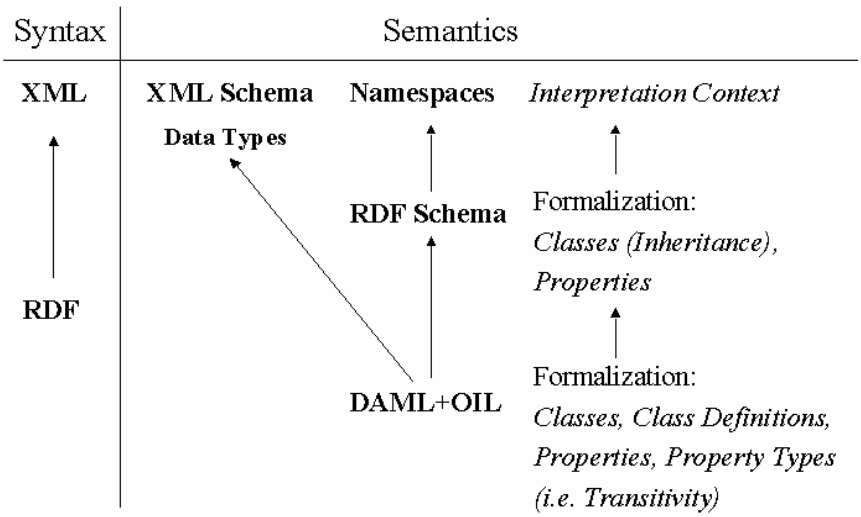


Figure 2-2: XML/RDF-based Knowledge Markup Languages

[12] http://www.w3.org/TR/rdf-schema/
[13] http://www.daml.org/2001/03/daml+oil-index
[14] http://www.w3.org/TR/owl-guide/
[15] http://www.w3.org/XML/Schema
[16] http://www.topicmaps.org/xtm/1.0/

For instance, defining the following XML statement to be in the `jobs` namespace ensures that the job of John Smith as a systems analyst is interpreted exactly as defined in this particular ontology.

```
<xmnls:jobs="http://www.jobs.org/daml+oil-jobs#">

<jobs:systems-analyst>John Smith</jobs:systems-analyst>, a
senior systems analyst with IBM, concluded that…
```

In this way, a semantic web agent will be able to identify John Smith as a systems analyst and look up additional knowledge on this concept in the `daml+oil-jobs` ontology, which it can access in a distributed fashion at the indicated namespace address.

2.2.2 Knowledge Organization and Access

2.2.2.1 Information Science

Within information science there is a long tradition of defining classification schemas (thesauri) for the organization and retrieval of available information in libraries and other archives. The already mentioned Dublin Core set of metadata results from this tradition. Also in future semantic web developments it is to be expected that tools and best practice techniques developed in information science will play a central role.

2.2.2.2 Machine Learning

Although classification and organization of knowledge is a highly intellectual and therefore human task, there is definitely a need for automatic support as the amount and complexity of the knowledge to be organized is rapidly growing. Machine learning approaches and tools are therefore needed to support the development, adaptation and use of ontologies on the semantic web.

2.2.2.3 Database Systems

The efficient organization of and access to knowledge largely depends on the availability of powerful database systems that can handle the storage and retrieval of large amounts of semantic objects on the semantic web, represented in RDF or other markup languages. Semantic objects may range from simple facts like “John Smith:systems-analyst” to complex objects such as instantiations of multi-agent negotiation protocols in e-commerce.

2.2.2.4 Language Technology

As for humans the use of language is still the most natural form of expressing knowledge, there will remain a need to transform this ambiguous medium into structured knowledge, to be accessed by agents and other web services on the semantic web. Therefore, language technology tools will be central in semantic web development in the following three areas: Knowledge Markup, Ontology Development, Intelligent Interfaces.

- Knowledge Markup: Turning the web into a semantic web implies widespread annotation of documents with ontology-based knowledge markup. Many of these

documents consist of free text in different languages, which can only be marked up in an efficient way by use of automatic, language technology tools.

- Ontology Development: Ontologies evolve rapidly over time and between different applications. Therefore, semi-automatic ontology learning that combines natural language processing (text mining, information extraction) with machine learning is essential for their efficient use.
- Intelligent Interfaces: Communication between humans and agents on the semantic web will be driven by natural language input, i.e. speech dialog. Obviously, language technology will be essential here in analysing user responses and in generation of appropriate synthetic responses by artificial agents.

3 Hybrid Web Page Annotation: RDF(S) Experiences *Antonio Pareja-Lora*

Following the guidelines of the Semantic Web initiative, as exposed all along this paper, much research has already been carried out by AI researchers on the semantic annotation of web pages. However, these researchers have been neglecting, somehow, the decades of work and the results obtained in the field of *Corpus Linguistics* on corpus annotation, not only in the semantic level, but also in other linguistic levels. These other linguistic levels, whilst not being intrinsically semantic, can also add some semantic information and help a computer understand a text.

This section presents some preliminary results from the *ContentWeb* project[17] (Aguado 2002) on how linguistic annotation can help computers understand the text contained in a document – a Semantic Web document, for example. Special efforts are being devoted in the aforementioned project to identifying complementarities between the semantic annotation models from AI and the annotations proposed by Corpus Linguistics, and combining both of them altogether; far from being irreconcilable, they may be considered complementary. Thus, an introduction to corpus annotation is offered in subsection 3.3.1. An example of the integration of both paradigms (AI's and Corpus Linguistics') in *ContentWeb –OntoTag–* is presented afterwards, in subsection 3.3.2. Finally, the main advantages and drawbacks of such a model are discussed in subsection 3.0.

3.1 Text Annotation in Corpus Linguistics

The idea of *text annotation* was originally developed in *Corpus Linguistics*. An **annotated corpus** "may be considered to be a repository of linguistic information [...] made explicit through concrete annotation" (McEnery and Wilson 2001). The benefit of such an annotation is clear: it makes retrieving and analysing information about what is contained in the corpus quicker and easier.

In EAGLES (1996a), a list of the main different **levels of linguistic annotation** can be found, namely: lemma, morphosyntactic, syntactic, semantic and discourse annotation. They are shown in Figure 3-1 (Annotation Level Pyramid), together with their corresponding tools (Linguistic Tool Stack) and applicable criteria, recommendations and guidelines (Linguistic Annotation Criteria Heap). A deeper analysis of these concepts than the one included below can be found in EsperOnto (2003).

[17] Supported by MCyT (Spanish Ministry of Science and Technology): "ContentWeb: Semantic Web Technologic Platform: Ontologies, Natural Language Analysis and E-Business" – TIC2001-2745. We would like to thank Guadalupe Aguado, Inmaculada Álvarez de Mon, Rosario Plaza and the LIA-Ontologies group for their collaboration in the research presented in this section.

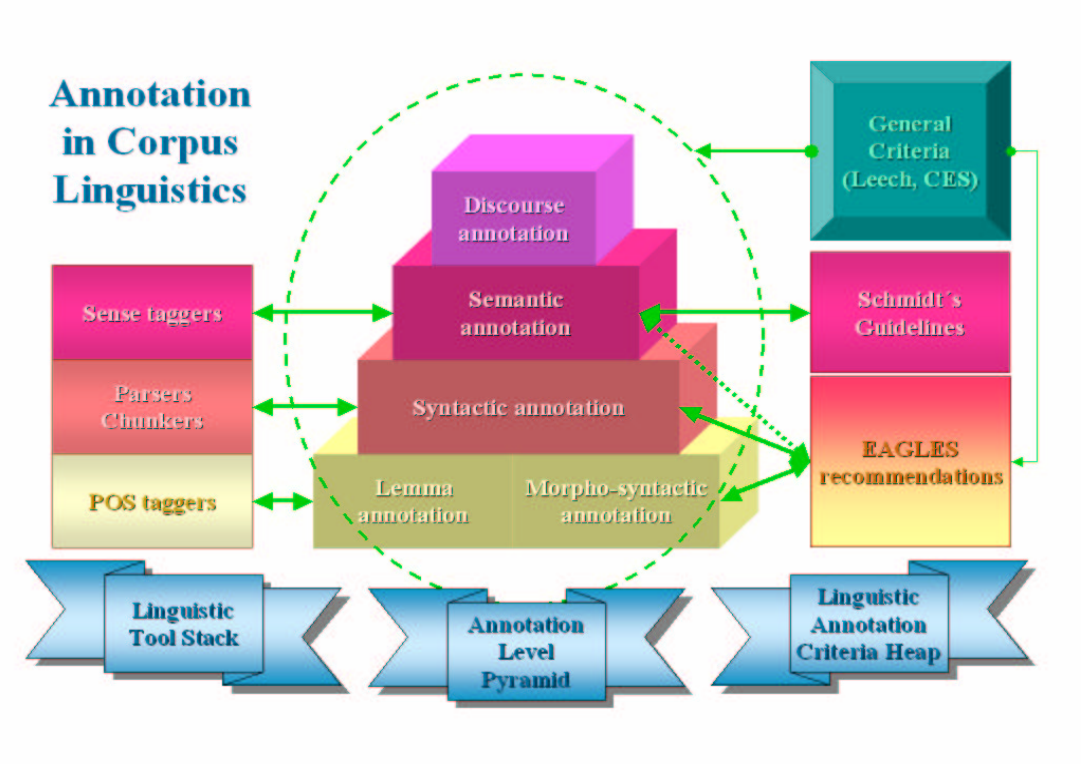


Figure 3-1: Annotation in Corpus Linguistics.

3.1.1 Lemma Annotation

Lemma annotation (lemmatisation) accompanies every word-token in a text with its **lemma**, that is, the form of the head word found when looking up that word in a dictionary. In English, lemma annotation may be considered redundant but, in more highly-inflected languages, such as Spanish, the ratio of word-forms per lemma makes lemma annotation a very valuable contribution to information extraction (Leech 1997). There are no specific guidelines for lemmatisation, so Leech's (1997) and CES (1999) criteria should be applied to this annotation level, which is usually carried out as a subtask by *POS-taggers*.

Once annotated, the lemma of a word can be linked to the concepts of an ontology and to the entries in a lexicon, functioning as a lexical semantic key for processing that word.

3.1.2 Morphosyntactic Annotation

Together with the syntactic annotation (next subsection), this is one of the most extended types in Corpus Linguistics. *Morphosyntactic annotation*, *part of speech annotation*, *POS tagging* or *grammatical tagging* is the annotation of the grammatical class (e.g. noun, verb, etc.) of each word-token in a text[18], together with (possibly) the annotation of its morphological analysis. As claimed in McEnery and Wilson (2001) POS information

[18] In other words, a POS tagging system holds the answer to the questions: a) How to divide the text into individual word tokens (words) b) How to choose a tagset (= a set of word categories to be applied to the word tokens) c) How to choose which tag is to be applied to which word (token).

establishes an essential foundation for further forms of analysis such as syntactic parsing and semantic field annotation and can be carried out automatically with an acceptable error rate by *POS-taggers*. EAGLES (1996a) offers a very valuable set of guidelines for this annotation level.

Disambiguation of homographs, identification of idiomatic word sequences and compounds or separation of contracted forms are some of the different irregularities an annotator must face at this level and the main contribution of this annotation level to the semantic analysis and processing of a document.

3.1.3 Syntactic Annotation

Once morphosyntactic categories in a text have been identified, *syntactic annotation* adds information about the higher-level syntactic relationships between these categories, which are determined, e.g., by means of a phrase-structure or dependency parse. Different parsing schemes are employed by different annotators; according to McEnery and Wilson (2001) these schemes differ in:

- The number of constituent types they employ (typically, the number of tags in the POS tagset).
- The way in which constituents are permitted to combine with one another.
- The grammar followed to parse and annotate the text.

Two kind of tools can be applied to syntactic annotation: *chunkers* (shallow parsing) and *parsers* (full, deeper parsing). The most interesting guidelines for this level can be found in EAGLES (1996b).

Some syntactic phenomena and properties can determine some –not necessary minor– changes in the semantics of an expression, such as the ordering of the words in a compound[19], the change of meaning of certain adjectives, in Spanish for example, when pre- and post- modifying a substantive[20] or the determination of PP-attachments.

3.1.4 Semantic Annotation

As asserted in McEnery and Wilson (2001), two broad types of semantic annotation may be identified, related to:

A. Semantic relationships between items[21] in the text (i.e., the agents or patients of particular actions). This type of annotation has scarcely begun to be applied.
B. Semantic features of words in a text, essentially the annotation of word senses in one form or another. There is no universal agreement in semantics about which features of words should be annotated[22].

---

[19] Compare, for example, "a Semantic Web page annotation model" and "a Semantic Web annotation model page".

[20] For instance, see the difference between "*un pobre hombre*" (an unlucky man) and "un hombre pobre" (a poor –having little money– man)

[21] The participants involved in an event, process or state described, the different phrases and clauses in the syntactic level, etc.

[22] See, for example, the controversies within the SENSEVAL initiative meetings (Kilgarriff 1998; Kilgarriff and Rosenzweig 2000).

As shown in Figure 3-1, the only tools available for automatic semantic annotation are sense taggers. Regarding the annotation criteria for this level, no EAGLES semantic corpus annotation standard has yet been published, although some preliminary recommendations on lexical semantic encoding have already been posted (EAGLES 1999); nevertheless, for the second type of semantic annotation enunciated, B, a set of reference criteria has been proposed by Schmidt (1988) for choosing or devising a corpus semantic field[23] annotation system. These criteria can be summarized as follows:

1. *It should make sense in linguistic or psycholinguistic terms.*
2. *It should be able to account exhaustively for the vocabulary in the corpus, not just for a part of it.*
3. *It should be sufficiently flexible* to allow for those emendations that are necessary for treating a different period, language, register or textbase.
4. *It should operate at an appropriate level of granularity (or delicacy of detail).*
5. *It should, where appropriate, possess a hierarchical structure.*
6. *It should conform to a standard, if one exists.*

The use of ontologies as a basis for a semantic annotation scheme fits perfectly and accomplishes the criteria posited by Schmidt. Clearly, the mostly hierarchical structure of ontologies fulfils by itself criterion (5) and, as a side effect, criteria (2) and (4), since the ontology can grow horizontally (scope extension) and vertically (specialisation). Criterion (3) is also satisfied by an ontology-based semantic annotation scheme, since we can always specialise the concepts in the ontology according to specific periods, languages, registers and textbases. Ontologies are, by definition, consensual and, thus, are closer to becoming a standard than many other knowledge models, as criteria (6) requires. Concerning criterion (1), quite a lot of groups developing ontologies are characterized by a strong interdisciplinary approach that combines Computer Science, Linguistics and (sometimes) Philosophy; then, an ontology-based approach should also make sense in linguistic terms. Hence, the ontologic and the linguistic points of view of the Semantic Web can be considered complementary and mutually enlightening.

3.1.5 Discourse Annotation

This is the least frequently developed kind of annotation, at least in corpora. Still, two main different kinds of approaches to annotation at this level can be found. *Stenström's approach* (McEnery and Wilson 2001) is based on what she called *discourse tags*, derived empirically from an initial analysis of a corpus subsample. These tags included categories such as 'apologies' (e.g. *sorry, excuse me*) or 'greetings' (e.g. *hello, good evening*) and were used to mark items whose role in the discourse dealt primarily with discourse management rather than with the propositional content. This first approach has never become widely used in corpus linguistics. Conversely, the pronoun reference or *anaphoric annotation* approach considers *cohesion*[24] a crucial factor in our understanding of the processes involved in reading, producing and comprehending discourse, which can

[23] A **semantic field** (sometimes also called a conceptual field, a semantic domain or a lexical domain) is a theoretical construct which groups together words that are related by virtue of their being connected – at some level of generality – with the same mental concept (Wilson and Thomas 1997).

[24] **Cohesion** (Halliday and Hasan 1976) is the vehicle by which elements in texts are interconnected through the use of pronouns, repetition, etc..

be considered the main contribution of this annotation level to the semantic interpretation of a document. A clear exponent of this approach is the UCREL discourse annotation scheme, together with many other anaphoric annotation schemes, such as De Rocha's, Gaizauskas and Humphries' and Botley's (Garside et al. 1997).

As shown in Figure 3-1, no automatic annotation tool or specific criteria has been developed for this level yet. Hence, as in the case of lemmatisation, Leech's (1997) and CES (1999) criteria should be applied when annotating at this level.

3.2 Hybrid Annotation: OntoTag

One of the four subtasks of *ContentWeb* (Aguado 2002) is the development of *OntoTag*, a model and environment for the hybrid –linguistic and ontological– annotation of web documents.

```
<contentWeb:FilmReview>
<contentWeb:text>Tras cinco años de espera y después de
        muchas habladurías, llega a nuestras pantallas la película
        más esperada de los últimos tiempos.</contentWeb:text>
</contentWeb:FilmReview>

<!-- Morpho-syntactic annotation excerpt -->
<morphAnnot:Word rdf:ID="1_16">
<morphAnnot:surface_form>la</morphAnnot:surface_form>
<morphAnnot:TradAnnot rdf:about="#trad_ann_info_1_16"/>
<morphAnnot:MBTAnnot rdf:about="#mbt_ann_info_1_16"/>
<morphAnnot:ConstrAnnot rdf:about="#constr_ann_info_1_16"/>
</morphAnnot:Word>
<morphAnnot:TradAnnot rdf:ID="trad_ann_info_1_16">
<trad:tag> ARTDFS </trad:tag>
<morphAnnot:lemma> el </morphAnnot:lemma>
</morphAnnot:TradAnnot>
<morphAnnot:MBTAnnot rdf:ID="mbt_ann_info_1_16">
<mbt:tag> TDFS0 </mbt:tag>
<morphAnnot:lemma> el </morphAnnot:lemma>
</morphAnnot:MBTAnnot>
<morphAnnot:ConstrAnnot rdf:ID="constr_ann_info_1_16">
<constr:tag> DET </constr:tag>
        <constr:genus>FEM</constr:genus>
        <constr:numerus>SG</constr:numerus>
<morphAnnot:lemma>la</morphAnnot:lemma>
<constr:synfunction>DN&gt;</constr:synfunction>
</morphAnnot:ConstrAnnot>
```

Figure 3-2: Morpho-syntactic annotation of the Spanish article "la".

Within the elaboration of OntoTag, a first exploration phase has been performed. A short example of this first phase, implemented in RDF(S), is presented below; the annotation of the Spanish sentence "*Tras cinco años de espera y después de muchas habladurías, llega*

a nuestras pantallas la película más esperada de los últimos tiempos."[25] at the first three levels is shown in Figure 3-2, Figure 3-3 and Figure 3-4.

At the morpho-syntactic level (Figure 3-2) every word or lexical token is given a different Uniform Resource Identifier (URI henceforth) and three possible categorisations are included, according to three different tagsets and systems we want to evaluate. Each tagset has been assigned a different class in the morphAnnot namespace: *TradAnnot* – CRATER tagset–, *MBTAnnot* –MBT (2002) tagset– and *ConstrAnnot* –Constraint Grammar, FDG (Tapanainen and Järvinen 1997) tagset–. For the sake of space, just the annotation of the article "*la*" has been included in the figure.

At the syntactic level (Figure 3-3) every syntactic relationship between morpho-syntactic items is given a new URI, so that it can be referenced in higher-level relationships or by other levels of the annotation model (i.e. <*synAnnot:Chunk rdf:ID= "1_510"*>). Also for the sake of space saving, just the annotation of the phrase "*la película más esperada de los últimos tiempos*" has been included in the figure.

```
<!-- Syntactic annotation excerpt -->
<synAnnot:Chunk rdf:ID="1_510">
    <synAnnot:synfunction>NP</synAnnot:synfunction>
    <synAnnot:hasChild rdf:about="#1_21">los</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_22">últimos</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_23">tiempos</synAnnot:hasChild>
</synAnnot:Chunk>
<synAnnot:Chunk rdf:ID="1_511">
    <synAnnot:synfunction>PP</synAnnot:synfunction>
    <synAnnot:hasChild rdf:about="#1_20">de</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_510"> los últimos tiempos
    </synAnnot:hasChild>
</synAnnot:Chunk>
<synAnnot:Chunk rdf:ID="1_512">
    <synAnnot:synfunction>AdjP</synAnnot:synfunction>
    <synAnnot:hasChild rdf:about="#1_18">más</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_19">esperada</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_511">de los últimos tiempos
    </synAnnot:hasChild>
</synAnnot:Chunk>
<synAnnot:Chunk rdf:ID="1_513">
    <synAnnot:synfunction>NP</synAnnot:synfunction>
    <synAnnot:hasChild rdf:about="#1_16">la</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_17">película</synAnnot:hasChild>
    <synAnnot:hasChild rdf:about="#1_512">más esperada de los últimos
    tiempos </synAnnot:hasChild>
</synAnnot:Chunk>
```

Figure 3-3: Syntactic annotation of the Spanish chunk "*la película más esperada de los últimos tiempos*".

At the semantic level (see Figure 3-4) some components of lower level annotations are tagged with semantic references to the concepts, attributes and relationships determined by our (domain) ontology, which is implemented in DAML+OIL. Further elements susceptible of semantic annotation are being sought and research is being done towards

[25] After five years of expectation and gossiping, here comes the most expected film for the time being.

their determination by the linguist team in our project. The pragmatic counterpart of OntoTag has not yet been tackled and, thus, this level is not included in the example.

```
<!-- Semantic annotation excerpt -->

<onto:PremiereEvent rdf:ID="_anon27">
        <semSynAnnot:includes rdf:about="#1_13">llega</semSynAnnot:includes>
        <semSynAnnot:includes rdf:about="#1_509">a nuestras pantallas</semSynAnnot:includes>
        <onto:hasFilm rdf:about="#_anon30"/>
</onto:PremiereEvent>

<onto:Film rdf:ID="_anon30">
    <semAnnot:includes rdf:about="#1_18">película</semAnnot:includes>
    <onto:comment rdf:about="#_anon40">
    <onto:comment rdf:about="#_anon41">
</onto:Film>

<onto:ControversialFilm rdf:ID="_anon40">
        <semSynAnnot:includes rdf:about="#1_506">después de muchas habladurías</semSynAnnot:includes>
</onto:ControversialFilm>

<onto:AwaitedFilm rdf:ID="_anon41">
    <semSynAnnot:includes rdf:about="#1_503">Tras cinco años de espera</semSynAnnot:includes>
    <semSynAnnot:includes rdf:about="#1_512">más esperada de los últimos tiempos</semSynAnnot:includes>
</onto:ControversialFilm>

<onto:Film rdf:about="#_anon30">
    <semSynAnnot:includes rdf:about="#3_507">El Señor de los Anillos</semSynAnnot:includes>
    <onto:filmTitle>El Señor de los Anillos</onto:filmTitle>
</onto:Film>
```

Figure 3-4: Semantic annotation of the Spanish sentence "Tras cinco años de espera y después de muchas habladurías, llega a nuestras pantallas la película más esperada de los últimos tiempos.".

3.3 Hybrid Annotation: Discussion

The integration of these two approaches (the linguistic and the ontological) into a hybrid annotation scheme entails many advantages for language engineering and AI applications. First of all, from the point of view of language resources:

- **Linguistic tools are reused**: the tools developed so far for lemmatisation, POS tagging and chunking/parsing can be exploited for the generation of the linguistic counterpart of the annotation.
- **Annotated documents** *are multi-purpose* and **can be reused**: once a document –web page– has been annotated, there is no need to POS-tag it, to parse it, etc., anymore, no matter what kind of processing it must go through afterwards (i.e. machine translation, information retrieval, information extraction, text mining, and so forth). Since parsing, for example, is a high time-consuming task, we can have an additional advantage, that is, reducing our overall Semantic Web page processing time.

The second main advantage is that the **meaning** of a page with explicit semantic annotation **can be reinforced by the meaning contribution provided by all of the linguistic levels**; semantic analysis can also benefit from the invaluable work done so far on the development of ontologies as conceptual and consensual hierarchical models, specially (UNSPSC 2002, RosettaNet 2002) in specific domains (i.e. e-commerce).

However, the main disadvantage lies in the limitations imposed by current technologies: the process of obtaining automatically compact, readable and verifiable pages is quite a hard task to be fully specified and delimited, but the work being done in our laboratory tries to bring some light upon it.

4 Using DAML for Representing Domain Specific Knowledge* *Barrett Bryant*

The Semantic Web envisioned by Berners-Lee, Hendler and Lassila (2001) requires software "agents" which operate across the web giving "intelligence" to web pages in the form of "services." Our view is that the Semantic Web may be thought of as a software system composed of many interoperating distributed heterogeneous software components. The software realizations of these components may span various component models but their interoperability is facilitated by their "knowledge" of a particular domain. Our research involves the formalization of these domains, or more precisely, development of the technology required to realize this formalization. We have applied this work to two problem areas: 1) formalization of domain knowledge to assist in understanding natural language requirements documents, and 2) formalization of domain models under which heterogeneous components may interoperate within a common understanding. Both types of formalizations have required a suitable knowledge representation language and toward this end we have experimented with XML (Decker et al. 2000), the eXtensible Markup Language, and DAML, the DARPA[26] Agent Markup Language (Hendler and McGuinness 2000). This section describes experiences with using DAML. We first briefly describe DAML and related semantic representation languages and then describe how we used DAML to represent domain-specific knowledge in our two research projects. Finally we conclude with some summary observations.

4.1 Semantic Representation Languages

XML has revolutionized the way in which the syntax of data may be represented for portability across a wide variety of platforms, languages and applications. Whereas XML has achieved syntactic interoperability, DAML strives for semantic interoperability. The development of DAML is due to a DARPA sponsored research project to extend XML with semantic relationships and the ability to express ontologies required for "understanding" such semantics. The emphasis is on describing semantic information in such a way that the aforementioned software agents will be able to seamlessly integrate Web pages, the software systems which are embedded in such pages, and the databases that such systems interact with. The original definition of DAML shared many characteristics with OIL (Fensel et al. 2001), the Ontology Inference Layer, especially an object-oriented type system. DAML+OIL represents a merger of these two approaches which has been proposed as a W3C[27] standard for representing ontologies. Bechhofer, Goble, and Horrocks (2001) point out that the DAML+OIL representations sometimes do

* This research is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number DAAD19-00-1-0350 and by the U. S. Office of Naval Research under award number N00014-01-1-0746.

[26] United States Defense Advanced Research Projects Agency (http://www.darpa.mil)
[27] World Wide Web Consortium (http://www.w3c.org)

not preserve all important information present in semantic models. A related effort to DAML+OIL is OWL (Smith et al. 2002), the Web Ontology Language, which provides semantics in the form of logical assertions constituting a knowledge base about classes and relationships between them that are omnipresent in Web applications.

Baclawski et al. (2001) use UML, the Unified Modeling Language, to represent ontologies. They found that there were many compatibilities between UML and DAML. Since UML is an OMG[28] standard for object-oriented modeling of software systems and hence has many associated tools, this work allows such tools to be used in representing semantic knowledge. This is also highly relevant to another OMG (2001) initiative, Model Driven Architecture (MDA), which proposes that software architecture be constructed around models developed within the context of standardized application domains.

We have used Two-Level Grammar (TLG) (Bryant and Lee 2002) to represent semantic information. The two levels of the grammar are a "meta-level" for describing the domains to be used at the "hyper-level," the logical rules over the domains. These grammars are encapsulated in classes and structured in an object-oriented manner for compatibility with objects and components, which they are used to describe the semantics of.

4.2 Representing Domain Specific Knowledge in DAML+OIL

Our research has used DAML+OIL for representing knowledge found in software requirements documents and domain knowledge about a particular application domain for which a software system is to be assembled from a collection of components. Each of these projects will be described in terms of a simple example, namely a bank account management system. Such a system typically consists of a database with account information, and a server exposing the database to clients such as customers using Web browsers, ATM's[29], or telephone interfaces. This system (and all other systems) may be viewed at two levels: 1) the software requirements for constructing the system, and 2) a collection of components which may be assembled to construct the system.

4.2.1 Software Requirements

A software requirements document typically describes the functionality of the system to be constructed, usually in natural language. The document may contain specific information about how the system should work, including non-functional aspects such as timing constraints, security, etc. A human reader may understand this information due to his/her knowledge of the domain, e.g. most software engineers would have a basic understanding of the functionality of a bank account management system. However, as software engineers aren't domain experts, it is often required that domain knowledge be explicitly described. Recent research in "generative programming" (Czarnecki and

[28] Object Management Group (http://www.omg.org)
[29] Automatic Teller Machines

Eisenecker 2000) has shown that much of the software development process may be automated (i.e., through generation of program components) when the domain is well specified. The domain knowledge required describes the relationship between components and other constraints which are usually presumed in requirements documents or too implicit to be extracted easily from the original documents. For example, the requirements for an ATM component in our banking system might say “the user inputs the 4 digit PIN number by pressing the buttons.” The fact that the set of the buttons is a component of the ATM machine is implicitly assumed and therefore not explicitly mentioned in the requirements documents. This kind of information is domain specific knowledge. The units of measurements, who passes what to whom, which synonyms of a word are used, what each acronym stands for, etc., are some of the examples of domain specific knowledge that can supplement software requirements documents.

We have used DAML to specify the domain knowledge required for a software system (Lee and Bryant 2002). The following examples show the use of DAML as domain knowledge for the ATM example. The `disjointUnionOf` notation in DAML can be used to list the subcomponents of a component, as in three data fields of an account:

```
<daml:Class rdf:ID="Account">
  <daml:disjointUnionOf
    rdf:parseType="daml:collection">
    <daml:Class rdf:ID="ID"/>
    <daml:Class rdf:ID="PIN"/>
    <daml:Class rdf:ID="Balance"/>
  </daml:disjointUnionOf>
</daml:Class>
```

where `rdf` stands for Resource Description Framework on which DAML and XML are built.

The `sameClassAs` definition in DAML may be used to indicate that the word `Machine` used in the ATM requirements is a synonym of the word `ATM` and that the word `ATM` stands for `Automatic Teller Machine`.

```
<daml:Class rdf:ID="Automatic_Teller_Machine">
  <daml:sameClassAs rdf:ID="Machine"/>
  <daml:sameClassAs rdf:ID="ATM"/>
</daml:Class>
```

Using `ObjectProperty` notation in DAML, the fact that a `Balance` is passed from a `Bank` to an `ATM` may be expressed as follows:

```
<daml:ObjectProperty rdf:ID="passBalance">
  <rdfs:domain rdf:ID="Bank"/>
  <rdfs:range rdf:ID="ATM"/>
</daml:ObjectProperty>
```

The data type or the measurement unit of a component may be expressed using `DatatypeProperty` notation in DAML as shown below for the type of `Amount`.

```
<daml:DatatypeProperty rdf:ID="Amount">
  <rdfs:range rdf:resource="http://www.w3.org/
   2000/10/XMLSchema#float"/>
</daml:DatatypeProperty>
```

In summary the precise formal semantics of DAML provides a very useful way to specify the domain specific knowledge explicitly. This knowledge is used as supplementary information for the conversion from a natural language requirements document into a formal specification in Two-Level Grammar, further details of which are described by Lee and Bryant (2003).

4.2.2 Semantic Web of Software Components

The UniFrame[30] (Raje et al. 2002) project aims to facilitate the construction of software systems via the integration of heterogeneous and distributed components deployed on the Web. Each software component will be accompanied by a formal specification indicating the functional and non-functional (also known as QoS – Quality of Service) properties of the component. The system assembler submits a query to a specialized search engine called a "head-hunter" which locates a possible set of components which satisfy the query using the formal specifications of the component. This process is facilitated by a formalized domain model of the application domain (e.g. bank account management systems) which each deployed component adheres to, as advocated in Model Driven Architecture (OMG, 2001). The formalized domain model is called a Generative Domain Model (GDM) since it contains rules to generate programs (in the sense of Generative Programming (Czarnecki and Eisenecker 2000)) which connect heterogeneous components, called "wrapper/glue code" to make a unified whole. The formalization aspects of this process are described further by Cao et al. (2002). For our bank account management system example, assume that on the Web are deployed various database components for storing bank account information, server-side components for accessing the database, and client-side components for interacting with the server. Each of these is accompanied by a formal specification of how the component adheres to the domain model, what technology it uses (e.g. CORBA[31], J2EE[32], or .NET[33]), what types of component interactions are expected, and the Quality of Service expected for the component. The system assembler indicates the domain of the system and what functionality is expected based upon the domain, as well as system QoS requirements. The UniFrame process assembles the various sets of compatible components into a

[30] Unified Framework for Seamless Integration of Heterogeneous Distributed Software Components (http://www.cs.iupui.edu/uniFrame)
[31] Common Object Request Broker Architecture (http://www.corba.org)
[32] Java 2 Enterprise Edition (http://java.sun.com)
[33] MicroSoft .NET Framework (http://www.microsoft.com/net)

collection of candidate software systems. Each software system is then tested with a set of test cases to validate that QoS requirements are met for the assembled system.

The GDM in our system is expressed in Two-Level Grammar with UML used to provide a visual representation. These may further be expressed in XML and related dialects, including DAML+OIL. However, we found that the expessiveness of TLG could not be conveniently captured in the current form of these notations. It is not difficult to represent the object-oriented type structure of TLG or the GDM ontological information in XML or DAML+OIL but the generation rules of TLG do not have such straightforward representations.

4.3 Summary

In conclusion, we found DAML+OIL to be a convenient representation for domain specific knowledge representations associated with software requirements documents, but not so suited for expressing generation rules in generative domain models. We did not investigate the use of DAML+OIL for expressing non-functional requirements such as Quality of Service but believe this would be an interesting exercise. We will continue to monitor the developments in languages for the Semantic Web for opportunities to integrate generation rule technologies and also to explore Two-Level Grammar as another possible language for the Semantic Web. Certainly TLG is able to capture semantic information pertaining to the integration of software components that is not currently available in Semantic Web languages.

5 Bringing Natural Language to the Semantic Web
Jimmy Lin

Because the ultimate purpose of the Semantic Web is to help users better locate, organize, and process content, we believe that it should be grounded in the information access method humans are most comfortable with—natural language. However, the Resource Description Framework (RDF), the foundation of the Semantic Web, was designed to be easily processed by computers, not humans. To render RDF friendlier to humans, we augment it with natural language annotations, or metadata written in everyday language. We argue that natural language annotations, parsed into computer-readable representations, are not only intuitive and effective, but can also accelerate the pace with which the Semantic Web is being adopted. We believe that our technology can facilitate a happy marriage between natural language technology and the Semantic Web vision.

5.1 The Semantic Web Vision

The vision of the Semantic Web is to convert existing Web information into a more machine-readable form, with the goal of making the Web more effective for users. This goal grew out of the recognition that although a wealth of information readily exists today in electronic form, this information lacks any machine-understandable semantics, and hence cannot be easily processed by computer systems. By adding semantics to existing Web data, e.g., this particular number is a price in Euros, we can create an environment that allows intelligent software agents to interoperate easily. However, we argue that such metadata alone is not enough to bring out the full potential of the Semantic Web.

Fundamentally, Semantic Web research is attempting to address the problem of information access: building systems that help users locate, collate, compare, and cross-reference content. As such, we believe that users should be able to access information using everyday language, and that the Semantic Web should be grounded in linguistically-motivated constructs. Natural language is intuitive, easy to use, rapidly deployable, and requires no specialized training. In our vision, the Semantic Web will be equally accessible by computers using specialized languages and interchange formats, and humans using natural language. The scenario of being able to ask a computer "When was the president of Taiwan born?" or "Find me the cheapest vacation package in the Bahamas this month" and getting back "just the right information" is very appealing.

Because the first step to building the Semantic Web is to transform existing sources (stored as HTML pages, in legacy databases, etc.) into a machine-understandable form (i.e., XML/RDF), it is sometimes at odds with a human-based natural language view of the world because formally and precisely defined ontologies necessary for computer comprehension may seem very unnatural to humans. Although the general framework of the Semantic Web includes provisions for natural language technology, the actual deployment of such technology remains largely unexplored. We fear that if greater

consideration is not given to the integration of language technologies at the onset, future solutions might be little more than ad-hoc patches.

We believe that the fundamental disjoint between the current Semantic Web vision and actual end users is that the Resource Description Framework (RDF), the standardized Semantic Web language for describing metadata, was meant for consumption by computers, not humans. Given this philosophy, how can we be sure that we're creating useful metadata? How can we be sure that our ontologies mirror the way users organize and think about content? Since the final beneficiary of the Semantic Web should be the user,[34] we advocate a human-centered organization of metadata grounded in natural language. We accomplish this by weaving *natural language annotations* directly into the Resource Description Framework.

5.2 Natural Language Annotations

Use of metadata is a common technique for rendering information fragments more tenable to processing by computer systems. Using natural language itself as metadata presents several additional advantages: it preserves human readability, allows for easy querying, and encourages non-expert users to engage in metadata creation. To this end, we have developed natural language annotations (Katz 1997), which are machine-parsable sentences and phrases that describe the content of various information segments. These annotations serve as metadata that describe the kinds of questions that a particular piece of knowledge is capable of answering.

To illustrate how this technology works, consider the following paragraph about Mars:

Mars has two small moons: Phobos and Deimos. Phobos (fear) and Deimos (panic) were named after the horses that pulled the chariot of the Greek war god Ares, the counterpart to the Roman god Mars…

This paragraph may be annotated with the following:

Mars' two moons
Phobos and Deimos are two moons orbiting Mars.

A question answering system would parse these two annotations and store the parsed structures (e.g., ternary expressions (Katz 1988)) with pointers back to the original information segment. To answer a question, the user query, parsed into the same type of structures, would be compared against the annotations stored in the knowledge base. Because this match would occur at the level of parsed representations, linguistically

[34] It is true that many parts of the Semantic Web will never have any contact with humans, and may be created only for the benefit of software agents, e.g., inventory management systems communicating with warehouses. For these applications, natural language may not be necessary. Nevertheless, a large fraction of the Semantic Web involves end users, where we believe natural language forms the best information access medium.

sophisticated machinery such as synonymy/hyponymy relations, ontologies, and structural transformation rules (e.g., "S-Rules" (Katz 1988; Katz and Levin 1988)) could be brought to bear on the matching process. If a match were found, the segment corresponding to the annotation would be returned to the user as the answer. Because sophisticated natural language processing could be invoked in matching questions with annotations, precision far beyond that of standard keyword-based information retrieval techniques could be achieved. In addition, a linguistically-based system allows for variations in user queries, e.g., alternate formulations, active/passive voice, nominalizations, etc.

In the above example, the natural language annotations would allow a question answering system to answer the following questions:

What satellites orbit Mars?
How many satellites orbit Mars?
What are the names of the Martian moons?

We have implemented the above technology in START[35] (Katz 1997), the first question answering system available on the World Wide Web. Since it came online in December 1993, START has engaged in exchanges with hundreds of thousands of users all over the world, supplying them with useful knowledge.

5.2.1 Natural Language Annotations for the Semantic Web

An important feature of our annotation concept is that any information segment can be annotated: not only text, but also images, multimedia, database queries, and even procedures. Attaching natural language annotations to RDF serves as the basis of our framework for bringing natural language access capabilities to the Semantic Web. Here we provide an illustrative example of this technology; more detailed descriptions of the underlying mechanisms can be found elsewhere (Katz *et al.* 2002).

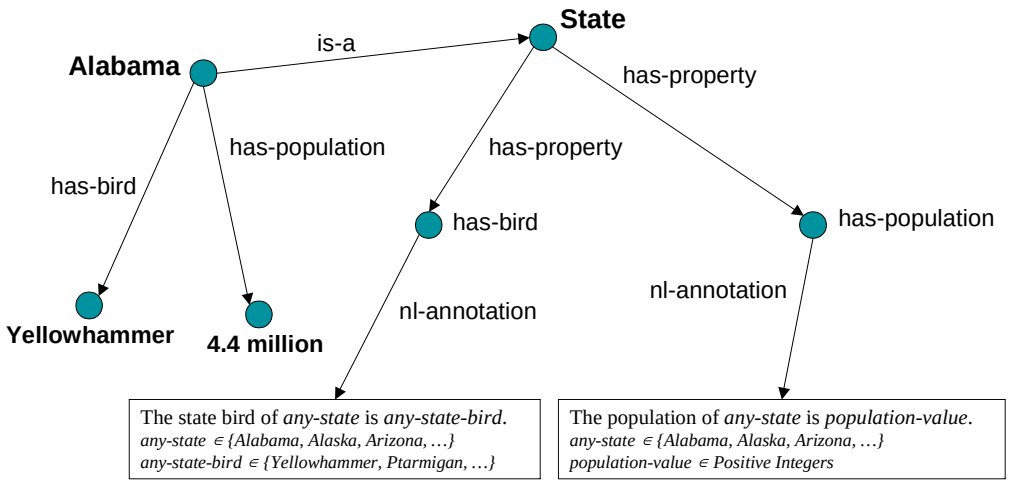
Suppose we want to answer the following "family" of questions about various attributes (e.g., state bird, state flower, state motto, population, area, etc.) of US states:

What is the state bird of California?
Tell me what the state motto of Massachusetts is.
Do you know Colorado's population?
What is the capital of Kentucky?

Fortunately, the data necessary to answer such questions can be easily found on the Web.[36] Assuming that this information has been structured into RDF,[37] our method of

[35] http://www.ai.mit.edu/projects/infolab
[36] http://www.50states.com/
[37] Currently, this must be accomplished manually, but in the future, information sources will be directly published in RDF.

bridging the Semantic Web and question answering can be conceptualized in the following ontology fragment:



In this simple ontology, a *state* "class" has a number of properties, e.g., *has-bird* and *has-population*. Attached to these properties are natural language annotations that describe, in stylized English, a linguistic realization of the information contained within the ontology. That is, the annotations are *parameterized* so that a single lexical item, e.g., *any-state,* can serve as a stand-in for a larger class of lexical items, e.g., all the 50 U.S. states. In effect, the annotation attached to the *has-bird* property is a shorthand way of specifying a large class of annotations: the state bird of Alabama is Yellowhammer, the state bird of Alaska is Ptarmigan, etc. One single parameterized annotation describes all individual instances of the *has-bird* property and relates all states to their respective state birds.

The question answering process proceeds as follows: A natural language question like "What is the population of Massachusetts?" is parsed and matched against the annotations. Because this match occurs on the level of syntactic representations, the matching mechanism can handle various formulations of the same question, e.g., dealing with lexical variations, semantic alternations, etc. The question matches the annotation of the *has-population* property, which triggers generation of the answer: the object of the *has-population* property of Massachusetts.

We have built a prototype implementing the technology described above (Katz *et al.* 2002) in the first Semantic Web question answering system that we are aware of. Although the system is currently limited in the types of questions that it can answer and the domain, we believe that the system is a proof of concept that demonstrates a viable method of integrating natural language techniques with the Semantic Web. By separating the knowledge, stored in RDF, from the natural language processing components, we can isolate ontology building from language engineering and vice versa, but still maintaining the connection between the two through natural language annotations.

Natural language annotations are part of our larger effort to build a uniform metadata framework for the Semantic Web (Karger *et al.* 2003). We wish to create the digital

equivalent of physical “sticky notes”, with the ability to attach annotations to anything and anywhere. Annotations written in natural language would complement other forms of more traditional, structured metadata such as author, creation date, subject, etc. Such a uniform framework would afford users the flexibility to query on multiple levels, using different sources of information. For example, consider a query such as “Show me mystery novels published within the last two years that other people have liked.” This request can only be fulfilled by searching over structured metadata (genre and publication date) and unstructured free text (user opinions). Our generalized annotation framework is built to accommodate exactly this type of user queries.

In our vision of the Semantic Web, natural language annotations would co-exist alongside other metadata under our generalized annotation framework. Metadata could be distributed (e.g., embedded directly into web pages) or centralized (e.g., stored in a common server); either way, a software agent would compile these digital “sticky notes” into a question answering system capable of providing natural language information access to users.

5.2.2 From Here to the Semantic Web

We believe that natural language annotations are not only an intuitive and helpful extension to the Semantic Web, but will assist in the deployment and adoption of the Semantic Web itself. The primary barrier to its creation is a classic chicken-and-egg problem: people will not spend extra time marking up their data unless they perceive a value for their efforts, and metadata will not be useful until a “critical mass” has been achieved. Although researchers have been focusing on technology to reduce barriers to entry (via authoring tools, for example), such initiative may not be sufficient to overcome the hurdles. As Hendler (2001) remarks, lowering markup cost isn't enough; for many users, the benefits of the Semantic Web must come for free. Semantic markup should be a by-product of normal computer use and there is no process of metadata creation that is easier and more intuitive than the use of natural language—users simply describe the contents of objects in everyday language. By divorcing the majority of users from the need to understand formal ontologies and a precisely defined vocabulary, we can dramatically lower barrier-of-entry, easing the transition into the Semantic Web vision.

Ultimately, let us not forget that the purpose of the Semantic Web is to benefit humans, not computers. The original idea was that instead of waiting for computers to become smart enough to understand human language, we should focus on the slightly less difficult problem of making human data more understandable to computers. To this end, the foundations of the Semantic Web are grounded in language and information access. However, to achieve interoperability and to facilitate interaction between software agents, we've had to sacrifice a lot of human understandability—precise ontologies and formally defined semantics are foreign concepts to the average user. By reintroducing natural language annotations and rendering the connection to human language explicit, we can achieve a satisfying middle ground between computer and human needs.

6 Conclusion: Language Technology and the Semantic Web *Nancy Ide*

In its broadest definition, the Semantic Web is intended to associate knowledge to web objects, whether they are documents, images, audio signals, and other media and processes. This knowledge will be represented in ontologies that define the relations among and properties of various bits of knowledge. While the bits of knowledge in the ontology may represent abstract concepts with no reference to a realization in any human language, it is unavoidable that the web objects with which they are associated will in the vast majority of cases consist of language data. Thus the role of language technology--which is the means by which we automatically discover meaning in language data or generate language data intended to represent specific meanings--in the Semantic Web is at the very least the means by which many of web objects will be identified, appropriately associated with ontological knowledge, and presented to the human user.

In terms of its relevance for language technology, it is useful to identify three different ways of thinking about the Semantic Web: (1) the Semantic Web as a long-term goal, realizing the vision of vast web of objects linked to a common ontology representing all knowledge, which in turn enables us and our intelligent agents to discover and manipulate these objects in sophisticated ("meaningful") ways; (2) the Semantic Web as it could be realized in the relatively near future using the technologies so far developed and under development intended to implement it; and (3) the means to actually *construct* the Semantic Web in both its near- and long-term realizations. Language processing technology will certainly play a seminal role in the construction of the Semantic Web by providing the tools required to automatically *identify* relevant objects in language data. Identification demands the usual range of language processing capabilities, depending on the kinds of objects under consideration: broad topical information can be identified using standard (often statistics-based) document retrieval techniques; information extraction can provide more specific information (e.g., more precise topical information, names, dates, etc.); detailed information about, for example, an event, its participants, causes and outcome may require deeper linguistic analysis. Similarly, machine learning techniques can be harnessed in the service of ontology-building for Semantic Web applications, and language/speech understanding and generation will be critical to the implementation of user-friendly interfaces. So, in the near future, we can assume that many of the activities central to language processing work will continue on in much the same way as they have done—except in one fundamental way.

Once identified by language processing applications, the relevant objects must be *annotated* to record the discovered information. Before the advent of the Semantic Web idea, annotation of language data typically involved identification of relevant segments (tokens, utterances, sentences, discourse units, etc.) and "labeling" them with some linguistic information for morpho-syntax, syntax, co-reference, parallel alignment, etc. Sometimes the labels are included in-line in the data; more recently, "stand-off" markup has been used to both delimit segments and link them to the relevant linguistic information in another document. Often, the labeling system is idiosyncratic to the data,

although recent standardization efforts have enabled some homogeneity in linguistic labeling among annotation projects.[38] Annotations at various linguistic levels are occasionally linked to specify relations among them (usually, constituent relations), but rarely, if ever, is an ontology of linguistic categories used in the background.

In the Semantic Web, annotation of objects is accomplished by associating an object with a category in an ontology, which in turn specifies its properties and relations to other categories. The big advantage of this idea, in addition to avoiding the duplication of information and ensuring a standard annotation scheme, is the ability to perform inferencing over the annotated data that enables the extraction of information that is not explicitly given. The notion that annotation will be accomplished via linkage to a common ontology of information brings up a number of questions about the ways in which language processing work can and should be accomplished in the immediate future.

The answer depends in part on what kinds of annotations we expect to be a part of the Semantic Web. It is possible to imagine that someday, as ontologies become increasingly rich, language processing capabilities become more robust, and computers become orders of magnitude faster, the only annotations that will be retained will be those at the highest levels, such as the representation of an event or state, together with its participants, their roles, etc., and lower-level linguistic information, such as syntactic structure, co-reference information, etc., may be computed on the fly and discarded once the really useful information is obtained. Eventually, it might be possible to generate much—possibly all--required information even at higher levels on the fly, and annotation of any kind will become obsolete.

However, for the foreseeable future, we can expect that intermediate annotations will be retained, and this brings up yet another question for language processing in the near term: should lower-level annotation types themselves be integrated into the Semantic Web technology? That is, should we be creating ontologies of *linguistic* categories together with their properties and relations, to be used in and by language processing applications? It may seem circular to utilize Semantic Web technologies to create ontologies to support the development of Semantic Web ontologies, but in fact the process is a bootstrap rather than a self-feeding loop. And it is a critical bootstrap, because we cannot expect semantic homogeneity to any degree at the higher levels if it is not achieved at the lower ones first.

The role of language technology in the Semantic Web is, then, twofold: first, established and evolving language processing techniques will play a crucial part in identifying objects to be integrated into the Semantic Web, developing the ontologies to support it, and enabling effective human-computer interaction that exploits the results. Second, it is up to the language processing community to employ the same technologies that will support the Semantic Web by encoding the requisite linguistic information in ontologies and exploiting inferencing capabilities in order to feed this effort. This second activity is in fact far more difficult than the first because it will demand, above all, an international collaborative effort to achieve it. This activity has barely begun, and it is not entirely clear how it can be accomplished. Some language processing researchers are developing

[38] However, note that in general the language processing community has resisted a common labeling scheme, for the good reason that different theoretical approaches cannot be represented with a common set.

ontological information to support Semantic Web applications (for example, the DAML effort sponsored by the U.S. Department of Defense) without full involvement of the international community, that are almost certainly bound to be domain-specific and ultimately unacceptable as off-the-shelf solutions. Other groups, such as the International Standards Organization Committee on Language Resources (ISO TC37 SC4), are attempting to work with the international community to achieve common standards by allowing for variation via formalized definitions of categories deviating from the stock of established norms; but even here, it is not clear how such deviations will be handled or tolerated by inferencing engines and other processing software. It will indeed be a very long road to achieve what is needed, but it is a road we must take with full awareness of not only the nature, but also the magnitude and complexity of the task.

Of course, at this point the Semantic Web is only a vision. Although it has been energetically embraced by much of the research community, its full realization is a very long way off. We are, in fact, in the stage where only the most fundamental groundwork for a Semantic Web is being laid, and the vision itself is so enormous and, to some extent, vague, that we cannot be sure exactly how the final product will turn out. Nonetheless, the Semantic Web seems to be a good idea (or at least the best idea we have at the moment), and we need to work towards achieving it even if along the way we find that the architecture has changed or the foundation needs major renovation. I cannot help but think of current work in language processing as “brain-building”, where we are attempting to cobble together a few hundred neurons here, a few hundred there, without much idea of how it all fits together in an interdependent network involving billions of such neurons that can accomplish language understanding at anything like the human level. The idea behind the Semantic Web, I believe, is one of those “intuitive leaps” that enabled us to have a suddenly clearer idea of how at least some of the pieces could be integrated, and this is likely the reason why so many have embraced and begun to pursue it. It’s a step, however modest, toward the eventual goal.

References

G. Aguado de Cea, I. Álvarez de Mon, R. Benjamins, J. Contreras, F. Martín, B. Navarrete, A. Pareja-Lora and R. Plaza-Arteche. 2003. *Esperonto Services IST-2001-34373 Deliverable D31 on Annotation Tools and Services*. http://www.esperonto.net/semanticportal/esperonto/ShowDeliverables.jsp

G. Aguado-de Cea, I. Álvarez de Mon, A. Pareja-Lora and R. Plaza-Arteche. 2002. OntoTag: A Semantic Web Page Linguistic Annotation Model. *Proceedings of the 2nd Workshop on NLP and XML (NLPXML-2002)*. COLING'2002. Taipei, Taiwan.

K. Baclawski, M. Kokar, P. Kogut, L. Hart, J. Smith, W. Holmes, J. Letkowski and M. Aronson. 2001. Extending UML to support ontology engineering for the Semantic Web. Proceedings of UML 2001, the Fourth International Conference on UML, 342-360.

S. Bechhofer, C. Goble and I. Horrocks. 2001. DAML+OIL is not enough. Proceedings of SWWS 2001, the First Semantic Web Working Symposium, 151-159.

T. Berners-Lee, J. Hendler and O. Lassila. 2001. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*. May, 2001. http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

B. R. Bryant and B.-S. Lee. 2002. Two-Level Grammar as an object-oriented requirements specification language. Proceedings of HICSS-35, the 35th Hawaii International Conference on System Sciences, http://www.hicss.hawaii.edu/HICSS_35/HICSSpapers/PDFdocuments/STDSL01.pdf.

F. Cao, B. R. Bryant, R. R. Raje, M. Auguston, A. M. Olson and C. C. Burt. 2002. Component specification and wrapper/glue code generation with Two-Level Grammar using domain specific knowledge. Proceedings of ICFEM 2002, the 4th International Conference on Formal Engineering Methods, 103-107.

CES. 1999. Corpus Encoding Standard. http://www.cs.vassar.edu/CES/

K. Czarnecki and U. W. Eisenecker. 2000. *Generative Programming: Methods, Tools, and Applications*. Reading, MA: Addison Wesley.

S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks. 2000. The Semantic Web: the roles of XML and RDF. *IEEE Internet Computing* 15: 63-74.

EAGLES. 1996a. *EAGLES: Recommendations for the Morphosyntactic Annotation of Corpora*. EAGLES Document EAG--TCWG—MAC/R.

EAGLES. 1996b. *EAGLES: Recommendations for the Syntactic Annotation of Corpora*. EAGLES Document EAG--TCWG—SASG/1.8.

EAGLES. 1999. *EAGLES LE3-4244: Preliminary Recommendations on Semantic Encoding*, Final Report.

D. Fensel, I. Horrocks, F. van Harmelen, D. McGuinness and P. F. Patel-Schneider. 2001. OIL: an ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems* 16: 38-45.

R. Garside, S. Fligelstone and S. Botley. 1997. Discourse Annotation: Anaphoric Relations in Corpora. *Corpus Annotation: Linguistic Information from Computer Text Corpora*, ed. by R. Garside, G. Leech and A. M. McEnery. London: Longman.

Y. Gil and V. Ratnaker. A Comparison of (Semantic) Markup Languages. In: *Proceedings of AAAI 2001*. (http://trellis.semanticweb.org/expect/web/semanticweb/comparison.html)

M. Halliday and R. Hasan. 1976 *Cohesion in English.* London: Longman.

J. Hendler. 2001. Agents and the SemanticWeb. *IEEE Intelligent Systems*, 16(2):30-37.

J. Hendler and D. L. McGuinness. 2000. The DARPA Agent Markup Language. *IEEE Intelligent Systems* 15: 67-73.

D. Karger, B. Katz, J. Lin and D. Quan. 2003. Sticky Notes for the Semantic Web. *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI 2003)*.

B. Katz. 1988. Using English for indexing and retrieving. *Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO '88)*.

B. Katz. 1997. Annotating the World Wide Web using natural language. *Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97).*

B. Katz and B. Levin. 1988. Exploiting lexical regularities in designing natural language systems. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING'88)*.

B. Katz, J. Lin and D. Quan. 2002. Natural Language Annotations for the Semantic Web. *Proceedings of the International Conference on Ontologies, Databases, and Application of Semantics (ODBASE 2002)*.

A. Kilgarriff. 1998. SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs. *Proceedings of LREC'98*, Granada, Spain, 581–588.

A. Kilgarriff and J. Rosenzweig. 2000. English SENSEVAL: Report and Results. *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'2000).* Athens, Greece.

B.-S. Lee and B. R. Bryant. 2002. Contextual processing and DAML for understanding software requirements specifications. Proceedings of COLING 2002, the 19th International Conference on Computational Linguistics, 516-522.

B.-S. Lee and B. R. Bryant. 2003. Applying XML technology for implementation of natural language specifications. *International Journal of Computer Systems, Science and Engineering*, 5: 3-24.

G. Leech. 1997. Introducing corpus annotation. *Corpus Annotation: Linguistic Information from Computer Text Corpora*, ed. by R. Garside, G. Leech and A. M. McEnery. London: Longman.

MBT. 2002. Visited on January 2002. http://ilk.kub.nl/~zavrel/tagtest.html

A. M. McEnery and A. Wilson. 2001. *Corpus Linguistics: An Introduction.* Edinburgh University Press, Edinburgh.

Object Management Group. 2001. *Model Driven Architecture: A Technical Perspective.* Needham, MA: Object Management Group (OMG). http://www.omg.org/mda.

R. R. Raje, M. Auguston, B. R. Bryant, A. M. Olson and C. C. Burt. 2002. A quality of service-based framework for creating distributed heterogeneous software components. *Concurrency and Computation: Practice and Experience*, 14 (12): 1009-1034.

RosettaNet. 2002. *RosettaNet: Lingua Franca for eBusiness.* Visited on April 2002. http://www.rosettanet.org/

K. M. Schmidt. (1988) Der Beitrag der begriffsorientierten Lexicographie zur systematischen Erfassung von Sprachwandel und das Begriffswörterbuch zur mhd. Epik. *Mittelhochdeutsches Wörterbuch in der Diskussion,* ed. by W. Bachofer. Tübingen: Max Niemeyer, 35–49.

M. K. Smith, D. McGuinness, R. Volz and C. Welty. 2002. *Web Ontology Language (OWL) Guide Version 1.0 (Working Draft)*. Cambridge, MA: World-Wide Web Consortium (W3C). http://www.w3.org/TR/2002/WD-owl-guide-20021104.

P. Tapanainen and T. Järvinen. 1997. A non-projective dependency parser. *Proceedings of the 5th conference on Applied Natural Language Processing*. Washington D.C.: Association for Computational Linguistics, 64–75.

UNSPSC. 2002. *Universal Standard Products and Services Classification (UNSPSC).* Visited on April 2002. http://www.unspsc.org/

A. Wilson and J. Thomas. 1997. Semantic Annotation. *Corpus Annotation: Linguistic Information from Computer Text Corpora*, ed. by R. Garside, G. Leech and A. M. McEnery. London: Longman.