Cardoso, J. (Ed.) "Semantic Web Services: Theory, Tools and Applications", Idea Group. Hard cover:978-1-59904-045-5, e-Book:978-1-59904-047-9, 2007.

The Syntactic and the Semantic Web

Jorge Cardoso

Department of Mathematics and Engineering
University of Madeira
9000-390 - Funchal
jcardoso@uma.pt

1 Motivation for the Semantic Web

The World Wide Web (WWW) was developed in 1989 at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland. It was Tim Berners-Lee who developed the first prototype of the World Wide Web intended to serve as an information system for physicists.

By the end of 1990, Tim Berners-Lee had written the first browser to retrieve and view hypertext documents and wrote the first Web server – the software, which stores Web pages on a computer for others to access. The system was originally developed to allow information sharing within internationally dispersed working groups. The original WWW consisted of documents (i.e. Web pages) and links between documents.

Browsers and Web server users grew. They became more and more attractive as an information sharing infrastructure. The Web became even more interesting as the amount of available information of every sort increased. A Web page can be accessed by a URL (Uniform Resource Locator) through the HyperText Transfer Protocol (HTTP) using a Web browser (e.g. Internet Explorer, Netscape, Mozilla, Safari).

Currently, the World Wide Web is primarily composed of documents written in HTML (Hyper Text Markup Language), a language that is useful for visual presentation. HTML is a set of "markup" symbols contained in a Web page intended for display on a Web browser. Most of the information on the Web is designed only for human consumption. Humans can read Web pages and understand them, but their inherent meaning is not shown in a way that allows their interpretation by computers.

The information on the Web can be defined in a way that can be used by computers not only for display purposes, but also for interoperability and integration between systems and applications. One way to enable machine-to-machine exchange and automated processing is to provide the information in such a way that computers can understand it. This is precisely the objective of the semantic Web – to make possible the processing of Web information by computers. "The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation." (Berners-Lee, Hendler et al. 2001). The next generation of the Web will combine existing Web technologies with knowledge representation formalisms (Grau 2004)

The Semantic Web was made through incremental changes, by bringing machine-readable descriptions to the data and documents already on the Web. Figure 1 illustrates the various developed technologies that made the concept of the Semantic Web

Cardoso, J. (Ed.) "Semantic Web Services: Theory, Tools and Applications", Idea Group. Hard cover:978-1-59904-045-5, e-Book:978-1-59904-047-9, 2007.

possible. As already stated, the Web was originally a vast set of static Web pages linked together. Many organizations still use static HTML files to deliver their information on the Web. However, to answer to the inherent dynamic nature of businesses, organizations are using dynamic publishing methods which offer great advantages over Web sites constructed from static HTML pages. Instead of a Web site comprising a collection of manually constructed HTML pages, server-side applications and database access techniques are used to dynamically create Web pages directly in response to requests from user browsers. This technique offers the opportunity to deliver Web content that is highly customized to the needs of individual users.

Nevertheless, the technologies available to dynamically create Web pages based on database information were insufficient for the requirements of organizations looking for application integration solutions. Businesses required their heterogeneous systems and applications to communicate in a transactional manner. The Extensible Markup Language (XML 2005) was one of most successful solutions developed to provide business-to-business integration. XML became a means of transmitting unstructured, semi-structured, and even structured data between systems, enhancing the integration of applications and businesses.

	Static	Dynamic	Syntax	Semantic
Encoding	HTML	+ RDBMS	+ XML	+ RDF/OWL
Creation	Manually	Generated by server-side applications	Generated by applications based on schema	Generated by applications based on models
Users	Humans	Humans	Humans and applications	Humans and applications
Paradigm	Browse	Create/Query/ Update	Integrate	Interoperate
Applications	Browsers	Browsers	Process Integration, EAI, BPMS, Workflows	Intelligent agents, Semantic engines

1995 2000 2005

Figure 1. Evolution of the Web

Unfortunately, XML-based solutions for applications and systems integration were not sufficient, since the data exchanged lacked an explicit description of its meaning. The integration of applications must also include a semantic integration. Semantic integration and interoperability is concerned with the use of explicit semantic descriptions to facilitate integration.

Currently the Web is undergoing evolution (as illustrated in Figure 2) and different approaches are being sought for solutions to adding semantics to Web resources. On the left side of Figure 2, a graph representation of the syntactic Web is given. Resources are linked together forming the Web. There is no distinction between resources or the links that connect resources. To give meaning to resources and links, new standards and languages are being investigated and developed. The rules and descriptive information made available by these languages allow the type of resources on the Web and the relationships between resources to be characterized individually and precisely, as illustrated on the right side of Figure 2.

Cardoso, J. (Ed.) "Semantic Web Services: Theory, Tools and Applications", Idea Group. Hard cover:978-1-59904-045-5, e-Book:978-1-59904-047-9, 2007.

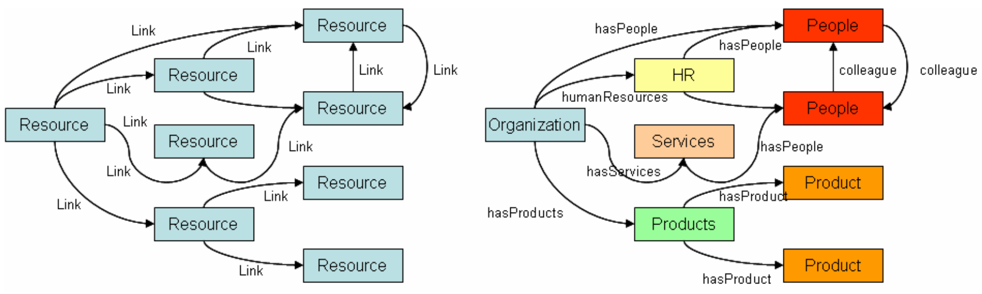

Figure 2. Evolution of the Web

Due to the widespread importance of integration and interoperability for intra- and inter-business processes, the research community has tackled this problem and developed semantic standards such as the Resource Description Framework (RDF) (RDF 2002) and the Web Ontology Language (OWL) (OWL 2004). RDF and OWL standards enable the Web to be a global infrastructure for sharing both documents and data, which make searching and reusing information easier and more reliable as well. RDF is a standard for creating descriptions of information, especially information available on the World Wide Web. What XML is for syntax, RDF is for semantics. The latter provides a clear set of rules for providing simple descriptive information. OWL provides a language for defining structured Web-based ontologies which allows a richer integration and interoperability of data among communities and domains.

2 The Visual and Syntactic Web

The World Wide Web composed of HTML documents can be characterized as a visual Web since documents are meant only to be displayed by Web browsers. In the visual Web, machines cannot understand the meaning of the information present in HTML pages, since they are mainly made up of ASCII codes and images. The visual Web prevents computers from automating information processing, integration, and interoperability.

With HTML documents, presentational metadata is used to assign information to the content and affect its presentation. Metadata is data about data and can be used to describe information about a resource. A resource can, for example, be a Web page, a document, an image, or a file. Examples of metadata that can be associated with a file include its title, subject, author, and size. Metadata mostly consists of a set of attribute value pairs that gives information about characteristics of a document. For example,

```
title = Semantic Web: Technologies and Applications
subject = Semantic Web
author = Jorge Cardoso
size = 336 Kbytes
```

In HTML pages, the content is marked-up with metadata. Specific tags are used to indicate the beginning and end of each element. For example, to specify that the title of the Web page is “Semantic Web: Technologies and Applications”, the text is marked-up using the tag <Title>. To inform the Web browser that “Motivation for the Semantic Web” is a heading, the text is marked-up as a heading element, using level-one <h1> heading tag such as:

```
<Title> Semantic Web: Technologies and Applications </Title>
<h1> Motivation for the Semantic Web </h1>
```

One restriction of HTML is that it is semantically limited. There is a lack of rich vocabulary of element types capable of capturing the meaning behind every piece of text. For example, Google search engine reads a significant number of the world's Web pages and allows users to type in keywords to find pages containing those keywords. There is no meaning associated to the keywords. Google only carries out simple matches between the keywords and the words in its database. The metadata of HTML is not considered when searching for a particular set of keywords. Even if Google would use HTML metadata to answer queries, the lack of semantics of HTML tags would most likely not improve the search.

On the other hand, the Syntactic Web is the collection of documents in the World Wide Web that contain data not just meant to be rendered by Web browsers, but also to be used for data integration and interoperability purposes. To be able to "understand" data, a computer needs metadata which will be provided by some kind of markup language. A widespread markup language is XML. With HTML the set of tags available to users is predefined and new tags cannot be added to the language. In contrast, XML is an extremely versatile markup language allowing users to be capable of creating new tags to add syntactic meaning to information.

In order to allow data integration, the meaning of XML document content is determined by agreements reached between the businesses that will be exchanging data. Agreements are usually defined using a standardized document, such as the Document Type Definition (DTD) (XML 2005) or the XML Schema Definition (XSD) (XMLSchema 2005) that specifies the structure and data elements of the messages exchanged. These agreements can then be used by applications to act on the data.

In a typical organization, business data is stored in many formats and across many systems and databases throughout the organization and with partner organizations. To partially solve integration problems, organizations have been using solutions such as XML to exchange or move business data between information systems. Prior to XML, an organization had to hardcode modules to retrieve data from data sources and construct a message to send to other applications. The adoption of XML accelerates the construction of systems that integrate distributed, heterogeneous data. The XML language allows the flexible coding and display of data, by using metadata to describe the structure of data (e.g. DTD or XSD).

The first step necessary to accomplish data integration using XML technologies consists of taking the raw data sources (text, spreadsheets, relational tables, etc) and converting them into well-formed XML documents. The next step is to analyze and document its structure by creating a DTD or XSD for each of the data sources.

One limitation of XML is that it can only define the syntax of documents. XML data does not include information which can be used to describe the meaning of the tags used. The following example illustrates an XML instance.

```
<student>
 <name> John Hall </name>
 <id> 669-33-2555 </id>
 <major> Philosophy </major>
</student>
```

In this example, the XML instance indicates there is a student named "John Hall". His <id> is "669-33-2555", but no information is provided about the meaning of an <id> or the meaning of the different fields that compose an <id>. Finally, the student's <major> is "Philosophy". No information is provided concerning the relationship of this <major> with the other majors that are given at the University John attends.

3 Unstructured, semi-structured, and structured data

Data breaks down into three broad categories (Figure 3): unstructured, semi-structured, and structured. Highly unstructured data comprises free-form documents or objects of arbitrary sizes and types. At the other end of the spectrum, structured data is what is typically found in databases. Every element of data has an assigned format and significance.

Unstructured data

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

Semi-structured data

```
<University>
 <Student ID="1">
   <Name>John</Name>
   <Age>18</Age>
   <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
   <Name>David</Name>
   <Age>31</Age>
   <Degree>Ph.D. </Degree>
 </Student>
….
</University>
```

Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Figure 3. Unstructured, semi-structured, and structured data

3.1 Unstructured data

Unstructured data is what we find in text, files, video, emails, reports, PowerPoint presentations, voice mail, office memos, and images. Data can be of any type and does not necessarily follow any format, rules, or sequence. For example, the data present on HTML Web pages is unstructured and irregular.

Unstructured data does not readily fit into structured databases except as binary large objects (BLOBs – Binary Large Objects). Although unstructured data can have some structure – e.g. e-mails have addressees, subjects, bodies, etc. and HTML Web pages have a set of predefined tags – the information is not stored in such a way that it will allow for easy classification, as the data is entered in electronic form.

3.2 Semi-structured data

Semi-structured data lies somewhere in between unstructured and structured data. Semi-structured data is data that has some structure, but is not rigidly structured. This type of data includes unstructured components arranged according to some pre-determined structure. Semi-structured data can be specified in such a way that it can be queried using general-purpose mechanisms.

Semi-structured data is organized into entities. Similar entities are grouped together, but entities in the same group may not have the same attributes. The order of attributes is not necessarily important and not all attributes may be required. The size and type of same attributes in a group may differ.

An example of semi-structured data is a Curriculum Vitae. One person may have a section of previous employments, another person may have a section on research experience, and another may have a section on teaching experience. We can also find a CV that contains two or more of these sections.

A very good example of a semi-structured formalism is XML which is a de facto standard for describing documents that is becoming the universal data exchange model on the Web and is being used for business-to-business transactions. XML supports the development of semi-structured documents that contain both metadata and formatted text. Metadata is specified using XML tags and defines the structure of documents. Without metadata, applications would not be able to understand and parse the content of

XML documents. Compared to HTML, XML provides explicit data structuring. XML uses DTD or XSD as schema definitions for the semi-structured data present in XML documents. Figure 3 shows the (semi) structure of an XML document containing students' records at a university.

3.3 Structured data

In contrast, structured data is very rigid and describes objects using strongly typed attributes, which are organized as records or tuples. All records have the same fields. Data is organized in entities and similar entities are grouped together using relations or classes. Entities in the same group have the same attributes. The descriptions for all the entities in a schema have the same defined format, predefined length, and follow the same order.

Structured data has been very popular since the early days of computing and many organizations rely on relational databases to maintain very large structured repositories. Recent systems, such as CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), and CMS (Content Management Systems) use structured data for their underlying data model.

4 Levels of semantics

As we have seen previously, semantics is the study of the meaning of signs, such as terms or words. Depending on the approaches, models, or methods used to add semantics to terms, different degrees of semantics can be achieved. In this section we identify and describe four representations that can be used to model and organize concepts to semantically describe terms, i.e. controlled vocabularies, taxonomies, thesaurus, and ontologies. These four model representations are illustrated in Figure 4.

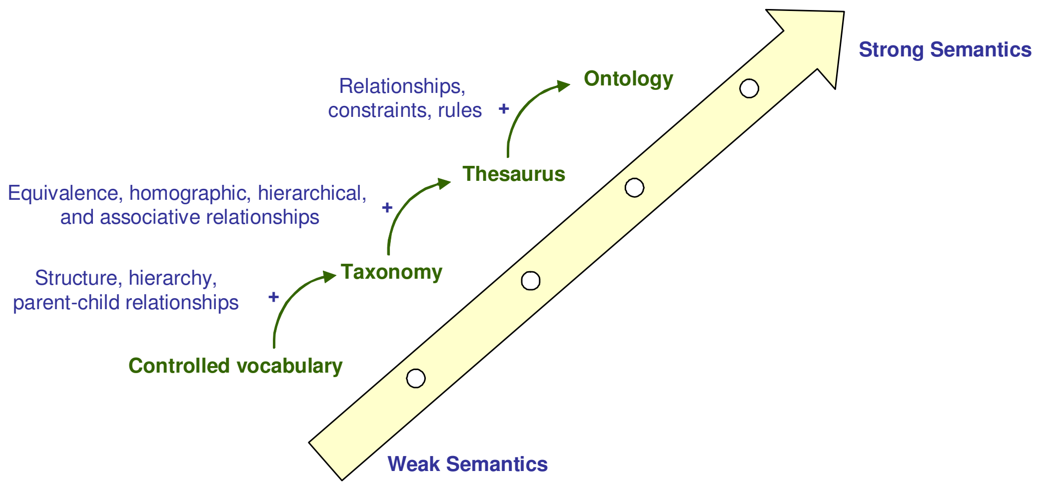

Figure 4. Levels of semantics

4.1 Controlled vocabularies

Controlled vocabularies are at the weaker end of the semantic spectrum. A controlled vocabulary is a list of terms (e.g., words, phrases, or notations) that have been enumerated explicitly. All terms in a controlled vocabulary should have an unambiguous, non-redundant definition. A controlled vocabulary is the simplest of all metadata methods and has been extensively used for classification. For example, Amazon.com has the following (Table 1) controlled vocabulary which can be selected by the user to search for products.

Books	Electronics	Cell Phones & Service
Popular Music	Camera & Photo	Outlet
Music Downloads	Software	Auctions
Classical Music	Tools & Hardware	zShops
DVD	Office Products	Everything Else
VHS	Magazines	Scientific Supplies
Apparel	Sports & Outdoors	Medical Supplies
Yellow Pages	Outdoor Living	Indust. Supplies
Restaurants	Kitchen	Automotive
Movie Showtimes	Jewelry & Watches	Home Furnishings
Toys	Beauty	Lifestyle
Baby	Gourmet Food Beta	Pet Toys
Computers	Musical Instruments	Arts & Hobbies
Video Games	Health/Personal Care	
	Travel	

Table 1. Controlled vocabulary used by Amazon.com

Controlled vocabularies limit choices to an agreed upon unambiguous set of terms. In cataloguing applications, users can be presented with list of terms from which they can pick the term to describe an item for cataloguing. The main objective of a controlling vocabulary is to prevent users from defining their own terms which can be ambiguous, meaningless, or misspelled.

4.2 Taxonomies

A taxonomy is a subject-based classification that arranges the terms in a controlled vocabulary into a hierarchy without doing anything further. The first users of taxonomies were biologists in the classification of organisms. They have employed this method to classify plants and animals according to a set of natural relationships. A taxonomy classifies terms in the shape of a hierarchy or tree. It describes a word by making explicit its relationship with other words. Figure 5 shows a taxonomy of merchandise that can be bought for a home.

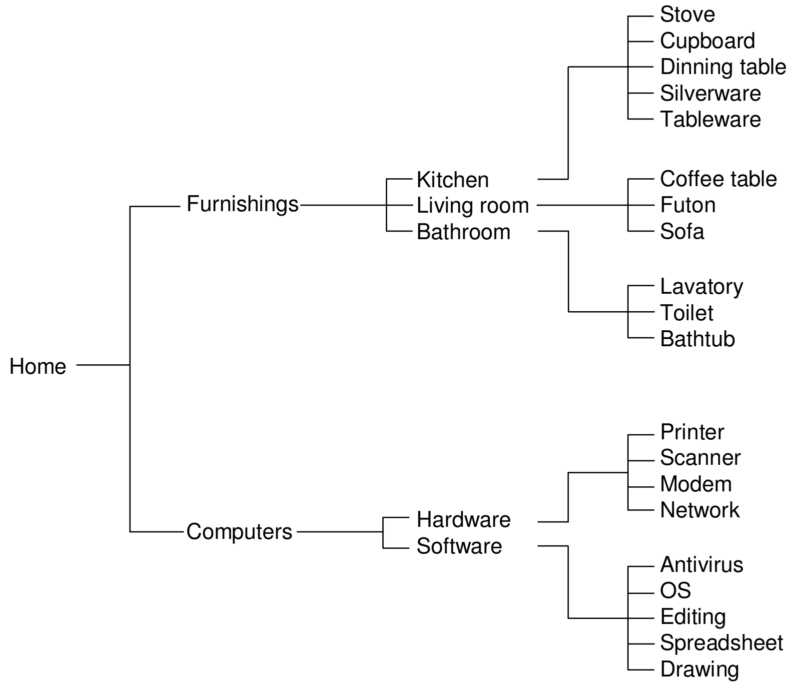

Figure 5: Example of a taxonomy

The hierarchy of a taxonomy contains parent-child relationships, such as “is subclass of” or “is superclass of”. A user or computer can comprehend the semantics of a word by analyzing the existing relationship between the word and the words around it in the hierarchy.

4.3 Thesaurus

A thesaurus is a networked collection of controlled vocabulary terms with conceptual relationships between terms. A thesaurus is an extension of a taxonomy by allowing terms to be arranged in a hierarchy and also allowing other statements and relationships to be made about the terms. A thesaurus can easily be converted into a taxonomy or controlled vocabulary. Of course, in such conversion, expressiveness and semantics are lost. Table 2 shows an example[1] of a thesaurus listing for the term *academic achievement*.

[1] http://fwrlibrary.troy.edu/1/dbhelp/dbhelp-psychology.htm

Relationship	Term
Used for	Grade point Average Scholastic Achievement School Achievement
Narrower than	Academic Overachievement Academic Underachievement College Academic Achievement Mathematics Achievement Reading Achievement Science Achievement
Broader than	Achievement
Related to	Academic Achievement Motivation Academic Achievement Prediction Academic Aptitude Academic Failure Academic Self Concept Education Educational Attainment Level School Graduation School Learning School Transition

Table 2. Example of a thesaurus listing for the term academic achievement

According to the National Information Standards Organization (NISO 2005), there are four different types of relationships that are used in a thesaurus: equivalence, homographic, hierarchical, and associative.

- Equivalence. An equivalence relation says that a term t_1 has the same or nearly the same meaning as a term t_2.
- Homographic. Two terms, t_1 and t_2, are called homographic if term t_1 is spelled the same way as a term t_2, but has a different meaning.
- Hierarchical. This relationship is based on the degrees or levels of "is subclass of" and "is superclass of" relationships. The former represents a class or a whole, and the latter refers to its members or parts.
- Associative. This relationship is used to link terms that are closely related in meaning semantically but not hierarchically. An example of an associative relationship can be as simple as "is related to" as in term t_1 "is related to" term t_2.

4.4 Ontologies

Ontologies are similar to taxonomies but use richer semantic relationships among terms and attributes, as well as strict rules about how to specify terms and relationships. In computer science, ontologies have emerged from the area of artificial intelligence. Ontologies have generally been associated with logical inferencing and recently have begun to be applied to the semantic Web.

An ontology is a shared conceptualization of the world. Ontologies consist of definitional aspects such as high-level schemas and assertional aspects such as entities, attributes, interrelationships between entities, domain vocabulary and factual knowledge – all connected in a semantic manner (Sheth 2003). Ontologies provide a common understanding of a particular domain. They allow the domain to be communicated between people, organizations, and application systems. Ontologies provide the specific tools to organize and provide a useful description of heterogeneous content.

In addition to the hierarchical relationship structure of typical taxonomies, ontologies enable cross-node horizontal relationships between entities, thus enabling easy modeling of real-world information requirements. Jasper and Uschold (1999) identify three major uses of ontologies:

1) to assist in communication between human beings
2) to achieve interoperability among software systems
3) to improve the design and the quality of software systems

An ontology is technically a model which looks very much like an ordinary object model in object-oriented programming. It consists of classes, inheritance, and properties (Fensel 2001). In many situations, ontologies are thought of as knowledge representation.

5 Semantic Web Architecture

The semantic Web identifies a set of technologies, tools, and standards which form the basic building blocks of an infrastructure to support the vision of the Web associated with meaning. The semantic Web architecture is composed of a series of standards organized into a certain structure that is an expression of their interrelationships. This architecture is often represented using a diagram first proposed by Tim Berners-Lee (Berners-Lee, Hendler et al. 2001). Figure 6 illustrates the different parts of the semantic Web architecture. It starts with the foundation of URIs and Unicode. On top of that we can find the syntactic interoperability layer in the form of XML, which in turn underlies RDF and RDF Schema (RDFS). Web ontology languages are built on top of RDF(S). The three last layers are the logic, proof, and trust, which have not been significantly explored. Some of the layers rely on the digital signature component to ensure security.

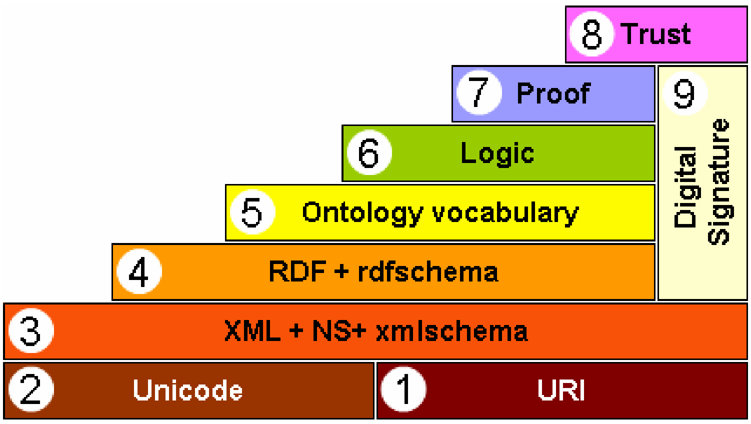

Figure 6: Semantic Web layered architecture (Berners-Lee, Hendler et al. 2001)

In the following sections we will briefly describe these layers. While the notions presented have been simplified, they provide a reasonable conceptualization of the various components of the semantic Web.

5.1 URI and Unicode

A Universal Resource Identifier (URI) is a formatted string that serves as a means of identifying abstract or physical resource. A URI can be further classified as a locator, a name, or both. Uniform Resource Locator (URL) refers to the subset of URI that identifies resources via a representation of their primary access mechanism. An Uniform Resource Name (URN) refers to the subset of URI that is required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable. For example,

- The URL http://dme.uma.pt/jcardoso/index.htm identifies the location from where a Web page can be retrieved
- The URN urn:isbn:3-540-24328-3 identifies a book using its ISBN

Unicode provides a unique number for every character, independently of the underlying platform, program, or language. Before the creation of unicode, there were various different encoding systems. The diverse encoding made the manipulation of data complex. Any given computer needed to support many different encodings. There was always the risk of encoding conflict, since two encodings could use the same number for two different characters, or use different numbers for the same character. Examples of older and well known encoding systems include ASCII and EBCDIC.

5.2 XML

XML is accepted as a standard for data interchange on the Web allowing the structuring of data on the Web but without communicating the meaning of the data. It is a language for semi-structured data and has been proposed as a solution for data integration problems, because it allows a flexible coding and display of data, by using metadata to describe the structure of data (using DTD or XSD).

In contrast to HTML, with XML it is possible to create new markup tags, such as <first_name>, which carry some semantics. However, from a computational perspective, a tag like <first_name> is very similar to the HTML tag <h1>. While XML is highly helpful for a syntactic interoperability and integration, it carries as much semantics as HTML. Nevertheless, XML solved many problems which have earlier been impossible to solve using HTML, i.e. data exchange and integration.

A well-formed XML document creates a balanced tree of nested sets of open and closed tags, each of which can include several attribute-value pairs. The following structure shows an example of an XML document identifying a 'Contact' resource. The document includes various metadata markup tags, such as <first_name>, <last_name>, and <email>, which provide various details about a contact.

```
<Contact contact_id="1234">
    <first_name> Jorge </first_name>
    <last_name> Cardoso </last_name>
    <organization> University of Madeira </organization>
    <email> jcardoso@uma.pt </email>
    <phone> +351 291 705 156 </phone>
</Contact>
```

While XML has gained much of the world's attention it is important to recognize that XML is simply a way of standardizing data formats. But from the point of view of semantic interoperability, XML has limitations. One significant aspect is that there is no way to recognize the semantics of a particular domain because XML aims at document structure and imposes no common interpretation of the data (Decker, Melnik et al.

2000). Another problem is that XML has a weak data model incapable of capturing semantics, relationships, or constraints. While it is possible to extend XML to incorporate rich metadata, XML does not allow for supporting automated interoperability of system without human involvement. Even though XML is simply a data-format standard, it is part of the set of technologies that constitute the foundations of the semantic Web.

5.3 RDF

At the top of XML, the World Wide Web Consortium (W3C) has developed the Resource Description Framework (RDF) (RDF 2002) language to standardize the definition and use of metadata. Therefore, XML and RDF each have their merits as a foundation for the semantic Web, but RDF provides more suitable mechanisms for developing ontology representation languages like OIL (Horrocks, Harmelen et al. 2001).

RDF uses XML and it is at the base of the semantic Web, so that all the other languages corresponding to the upper layers are built on top of it. RDF is a formal data model for machine understandable metadata used to provide standard descriptions of Web resources. By providing a standard way of referring to metadata elements, specific metadata element names, and actual metadata content, RDF builds standards for XML applications so that they can interoperate and intercommunicate more easily, facilitating data and system integration and interoperability. At first glance it may seem that RDF is very similar to XML, but a closer analysis reveals that they are conceptually different. If we model the information present in a RDF model using XML, human readers would probably be able to infer the underlying semantic structure, but general purpose applications would not.

RDF is a simple general-purpose metadata language for representing information in the Web and provides a model for describing and creating relationships between resources. A resource can be a thing such as a person, a song, or a Web page. With RDF it is possible to add pre-defined modeling primitives for expressing semantics of data to a document without making any assumptions about the structure of the document. RDF defines a resource as any object that is uniquely identifiable by a Uniform Resource Identifier (URI). Resources have properties associated to them. Properties are identified by property-types, and property-types have corresponding values. Property-types express the relationships of values associated with resources. The basic structure of RDF is very simple and basically uses RDF triples in the form of subject, predicate, object.

- subject: a thing identified by its URL
- predicate: the type of metadata, also identified by a URL (also called the property)
- object: the value of this type of metadata

RDF has a very limited set of syntactic constructs, no other constructs except for triples is allowed. Every RDF document is equivalent to an unordered set of triples. The example from Figure 7 describes the following statement using a RDF triple:

"Jorge Cardoso created the Jorge Cardoso Home Page."

Cardoso, J. (Ed.) "Semantic Web Services: Theory, Tools and Applications", Idea Group. Hard cover:978-1-59904-045-5, e-Book:978-1-59904-047-9, 2007.

The ‘Jorge Cardoso Home Page’ is a resource. This resource has a URI: http://dme.uma.pt/jcardoso/ and It has a property, ‘creator’, with the value ‘Jorge Cardoso’.

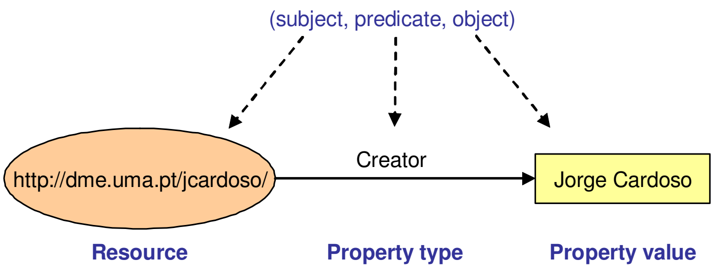

Figure 7. Graphic Representation of a RDF statement

The graphic representation of Figure 7 is expressed in RDF with the following statements:

```
<? xml version="1.0" ?>
<RDF xmlns = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
     xmlns:DC = "http://dublincore.org/2003/03/24/dces#">

  <Description about = "http://dme.uma.pt/jcardoso/">
    <DC:Creator> Jorge Cardoso </DC:Creator>
  </Description>
</RDF>
```

The first lines of this example use namespaces to explicitly define the meaning of the notions that are used. The first namespace xmlns:rdf=”http://w3.org/TR/1999/PR-rdf-syntax-19990105#” refers to the document describing the syntax of RDF. The second namespace http://dublincore.org/2003/03/24/dces# refers to the description of the Dublin Core (DC), a basic ontology about authors and publications.

The Dublin Core (DC 2005) is a fifteen element metadata set that was originally developed to improve resource discovery on the Web. To this end, the DC elements were primarily intended to describe Web-based documents. Examples of the Dublin Core metadata include:

- Title – the title of the resource
- Subject – simple keywords or terms taken from a list of subject headings
- Description – a description or abstract
- Creator – the person or organization primarily responsible for the intellectual content of the resource
- Publisher – the publisher
- Contributor – a secondary contributor to the intellectual content of the resource

The following example shows a more real and complete scenario using the DC metadata. It can be observed that more than one predicate-value pair can be indicated for a resource. Basically, it expresses that the resource ‘http://dme.uma.pt/jcardoso’ has the title ‘Jorge Cardoso Web Page’, its subject is ‘Home Page’, and was created by ‘Jorge Cardoso’.

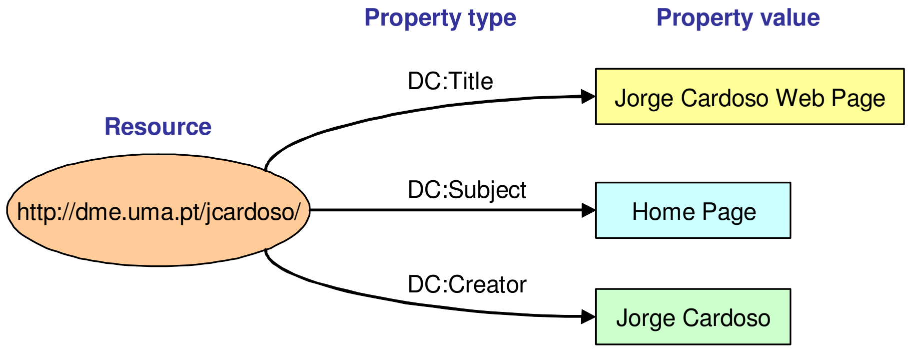

Figure 8. Graphic Representation of a RDF statement

The graphic representation of Figure 8 is expressed in RDF using the DC namespace with the following statements:

```
<? xml version="1.0" ?>
<RDF xmlns = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
     xmlns:DC = " http://dublincore.org/2003/03/24/dces#">

  <Description about = "http://dme.uma.pt/jcardoso/" >
    <DC:Title> Jorge Cardoso Home Page </DC:Title>
    <DC:Creator> Jorge Cardoso </DC:Creator>
    <DC:Date> 2005-07-23 </DC:Date>
  </Description>
</RDF>
```

Very good examples of real world systems that use RDF are the applications developed under the Mozilla project (Mozilla 2005). Mozilla software applications use various different pieces of structured data, such as bookmarks, file systems, documents, and sitemaps. The creation, access, query, and manipulation code for these resources is completely independent. While the code is completely independent, there is considerable overlap in the data model used by all these different structures. Therefore, Mozilla uses RDF to build a common data model shared by various applications, such as viewers, editors, and query mechanisms.

5.4 RDF Schema

The RDF Schema (RDFS 2004) provides a type system for RDF. The RDFS is technologically advanced compared to RDF since it provides a way of building an object model from which the actual data is referenced and which tells us what things really mean.

Briefly, the RDF Schema (RDFS) allows users to define resources with classes, properties, and values. The concept of RDF class is similar to the concept of class in object-oriented programming languages such as Java and C++. A class is a structure of similar things and inheritance is allowed. This allows resources to be defined as instances of classes, and subclasses of classes. For example, the RDF Schema allows resources to be defined as instances of one or more classes. In addition, it allows classes to be organized in a hierarchical fashion. For example the class First_Line_Manager might be defined as a subclass of Manager which is a subclass of Staff, meaning that any resource which is in class Staff is also implicitly in class First_Line_Manager as well.

An RDFS property can be viewed as an attribute of a class. RDFS properties may inherit from other properties, and domain and range constraints can be applied to focus their use. For example, a domain constraint is used to limit what class or classes a specific property may have and a range constraint is used to limit its possible values. With these extensions, RDFS comes closer to existing ontology languages. RDFS is used to declare vocabularies, the sets of semantics property-types defined by a particular community. As with RDF, the XML namespace mechanism serves to identify RDFS. The following statements illustrate a very simple example of RDFS where classes and inheritance are used.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base= "http://www.hr.com/humanresources#">

  <rdf:Description rdf:ID="staff">   class
   <rdf:type
   rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>                 subclass of

  <rdf:Description rdf:ID="manager"> class
   <rdf:type
   rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf rdf:resource="#staff"/>
  </rdf:Description>

</rdf:RDF>
```

The rdfs:Class is similar to the notion of a class in object-oriented programming languages. When a schema defines a new class, the resource representing that class must have an rdf:type property whose value is the resource rdfs:Class. Anything described by RDF expressions is called a resource and is considered to be an instance of the class rdfs:Resource. Other elements of RDFS are illustrated in Figure 9 and described bellow.

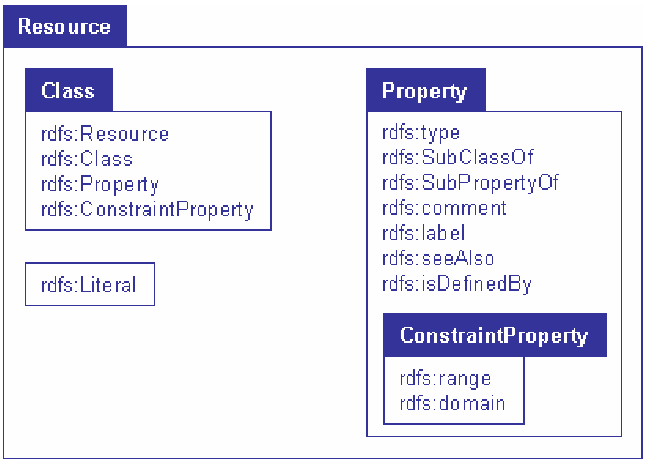

Figure 9. Relationships between the concepts of RDF Schema

- rdfs:Datatype is the class of data types and defines the allowed data types.
- rdfs:Literal is the class of literal values such as strings and integers.
- rdfs:subClassOf is a transitive property that specifies a subset-superset relation between classes.

- rdfs:subPropertyOf is an instance of rdf:Property used to specify that one property is a specialization of another.
- rdfs:comment is a human-readable description of a resource.
- rdfs:label is a human-readable version of a resource name and it can only be a string literal.
- rdfs:seeAlso specifies a resource that might provide additional information about the subject resource.
- rdfs:isDefinedBy is a subproperty of rdfs:seeAlso and indicates the resource defining the subject resource.
- rdfs:member is a super-property of all the container membership properties
- rdfs:range indicates the classes that the values of a property must be members of.
- rdfs:domain indicates the classes on whose member a property can be used.
- rdfs:Container is a collection of resources.
- rdfs:ContainerMemberShipProperty is a class that is used to state that a resource is a member of a container.

5.5 Ontologies

An ontology is an agreed vocabulary that provides a set of well-founded constructs to build meaningful higher level knowledge for specifying the semantics of terminology systems in a well defined and unambiguous manner. For a particular domain, an ontology represents a richer language for providing more complex constraints on the types of resources and their properties. Compared to a taxonomy, ontologies enhance the semantics of terms by providing richer relationships between the terms of a vocabulary. Ontologies are usually expressed in a logic-based language, so that detailed and meaningful distinctions can be made among the classes, properties, and relations.

Ontologies can be used to increase communication either between humans and computers. The three major uses of ontologies (Jasper and Uschold 1999) are:

- To assist in communication between humans.
- To achieve interoperability and communication among software systems.
- To improve the design and the quality of software systems.

In the previous sections, we have established that RDF/S was one of the base models and syntax for the semantic Web. On the top of the RDF/S layer it is possible to define more powerful languages to describe semantics. The most prominent markup language for publishing and sharing data using ontologies on the Internet is the Web Ontology Language (OWL 2004). Web Ontology Language (OWL) is a vocabulary extension of RDF and is derived from the DAML+OIL language (DAML 2001), with the objective of facilitating a better machine interpretability of Web content than that supported by XML and RDF. OWL adds a layer of expressive power to RDF/S, providing powerful mechanisms for defining complex conceptual structures, and formally describes the semantics of classes and properties used in Web resources using, most commonly, a logical formalism known as Description Logic (DL 2005).

Let's analyze some of the limitations of RDF/S to identify the extensions that are needed:

1. RDF/S cannot express equivalence between properties. This is important to be able to express the equivalence of ontological concepts developed by separate working groups.

2. RDF/S does not have the capability of expressing the uniqueness and the cardinality of properties. In some cases, it may be necessary to express that a particular property value may have only one value in a particular class instance.
3. RDF/S can express the values of a particular property but cannot express that this is a closed set. For example, an enumeration for the values for the gender of a person should have only two values: male and female.
4. RDF/S cannot express disjointedness. For example, the gender of a person can be male or female. While it is possible in RDF/S to express that John is a male and Julie a female, there is no way of saying that John is not a female and Julie is not a male.
5. RDF/S cannot express the concept of unions and intersections of classes. This allows the creation of new classes that are composed of other classes. For example, the class "staff" might be the union of the classes "CEO", "manager" and "clerk". The class "staff" may also be described as the intersection of the classes "person" and "organization employee".

Let us see a more detailed example of RDF/S limitations. Consider the sentence:

> *"There are three people responsible for the Web resource 'Jorge Cardoso Home Page' created in 23 July 2005: Web designer, editor, and graphic designer. Each has distinct roles and responsibilities."*

Using RDF/S we could try to model this statement in the following way:

```
<? xml version="1.0" ?>
<RDF xmlns = "http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
     xmlns:DC = " http://dublincore.org/2003/03/24/dces#">
     xmlns:S = " http://hr.org/2005/01/14/hr#">

  <Description about = "http://dme.uma.pt/jcardoso/" >
    <DC:Title> Jorge Cardoso Home Page </DC:Title>
    <DC:Creator> Jorge Cardoso </DC:Creator>
    <DC:Date> 2005-07-23 </DC:Date>
    <S:Administrator>
      <rdf:Bag>
        <rdf:li resource="Web designer"/>
        <rdf:li resource="Editor"/>
        <rdf:li resource="Graphic designer"/>
      </rdf:Bag>
    </S:Administrator>
  </Description>
</RDF>
```

In this example we have used the bag container model. In RDF, the container model is restricted to three components: bags, sequence, and alternative. Bags are an unordered list of resources or literals. A sequence is an ordered list of resources or literals. Finally, alternative is a list of resources or literals that represent alternatives for the (single) value of a property.

Using any of the three different relationships in RDF, we are only able to explain the information about the resources, but we cannot explain the second part of our statement, i.e. "Each has distinct roles and responsibilities."

Using OWL, we can represent the knowledge associated with the second part of our statement as shown bellow.

```
<owl:AllDifferent>
 <owl:distinctMembers rdf:parse Type="Collection">
   <admin:Administrator rdf:about="#Web designer"/>
   <admin:Administrator rdf:about="#Editor"/>
   <admin:Administrator rdf:about="#Graphic designer"/>
</owl:distinctMembers>
</owl:AllDifferent>
```

The owl:AllDifferent element is a built-in OWL class, for which the property owl:distinctMembers is defined, which links an instance of owl:AllDifferent to a list of individuals. The intended meaning of such a statement is that the individuals in the list are all different from each other. This OWL representation can express that the three administrators (Web designer, Editor, and Graphic designer) have distinct roles. Such semantics cannot be expressed using RDF, RDFS, or XML.

5.6 Logic, Proof, and Trust

The purpose of this layer is to provide similar features to the ones that can be found in First Order Logic (FOL). The idea is to state any logical principle and allow the computer to reason by inference using these principles. For example, a university may decide that if a student has a GPA higher than 3.8, then he will receive a merit scholarship. A logic program can use this rule to make a simple deduction: “David has a GPA of 3.9, therefore he will be a recipient of a merit scholarship.”

Inference engines, also called reasoners, are software applications that derive new facts or associations from existing information. Inference and inference rules allow for deriving new data from data that is already known. Thus, new pieces of knowledge can be added based on previous ones. By creating a model of the information and relationships, we enable reasoners to draw logical conclusions based on the model. The use of inference engines in the semantic Web allows applications to inquire why a particular conclusion has been reached, i.e. semantic applications can give proof of their conclusions. Proof traces or explains the steps involved in logical reasoning.

For example, with OWL it is possible to make inferences based on the associations represented in the models, which primarily means inferring transitive relationships. Nowadays, many inference engines are available. For instance:

- Jena reasoner – Jena includes a generic rule based inference engine together with configured rule sets for RDFS and for OWL. It is an open source Java framework for writing semantic Web applications developed by HP Labs (Jena 2005).
- Jess – Using Jess (Gandon and Sadeh 2003) it is possible to build Java software that has the capacity to “reason” using knowledge supplied in the form of declarative rules. Jess has a small footprint and it is one of the fastest rule engines available. It was developed at Carnegie Melon University.
- SWI-Prolog Semantic Web Library – Prolog is a natural language for working with RDF and OWL. The developers of SWI-Prolog have created a toolkit for creating and editing RDF and OWL applications, as well as a reasoning package (Wielemaker 2005).
- FaCT++ – This system is a Description Logic reasoner, which is a re-implementation of the FaCT reasoner. It allows reasoning with the OWL language (FaCT 2005).

Trust is the top layer of the Semantic Web architecture. This layer provides authentication of identity and evidence of the trustworthiness of data and services. While the other layers of the semantic Web stack have received a fair amount of attention, no significant research has been carried out in the context of this layer. The idea is to allow people to ask questions concerning the trustworthiness of the information on the Web. Possible scenarios for the trust layer include the possibility to make statements such as "I trust all information from http://dme.uma.pt/jcardoso, but I don't trust anything from http://www.internetsite.com".

6 Applications of the semantic Web

Even though the Semantic Web is still in its infancy, there are already applications and tools that use this conceptual approach to build semantic Web based systems. The intention of this section is to present the state of the art of the applications that use semantics and ontologies. We describe various applications ranging from the use of semantic Web services, semantic integration of tourism information sources, and semantic digital libraries to the development of bioinformatics ontologies.

Semantic Web services. Web services are modular, self-describing, self-contained applications that are accessible over the Internet (Curbera, Nagy et al. 2001). Currently, Web services are described using the Web Services Description Language (Chinnici, Gudgin et al. 2003), which provide operational information. Although the Web Services Description Language (WSDL) does not contain semantic descriptions, it specifies the structure of message components using XML Schema constructs. One solution to create semantic Web services is by mapping concepts in a Web service description (WSDL specification) to ontological concepts. The WSDL elements that can be marked up with metadata are operations, messages, and preconditions and effects, since all the elements are explicitly declared in a WSDL description.

Semantic Tourism Information Systems: Dynamic packaging technology helps online travel customers to build and book vacations. It can be described as the ability for a customer to put together elements of a (vacation) trip including flights, hotels, car rentals, local tours and tickets to theatre and sporting events. The package that is created is handled seamlessly as one transaction and requires only one payment from the consumer, hiding the pricing of individual components. So far, the travel industry has concentrated its efforts on developing open specification messages, based on XML, to ensure that messages can flow between industry segments as easily as within. For example, the OpenTravel Alliance (OTA 2004) is an organization pioneering the development and use of specifications that support e-business among all segments of the travel industry. It has produced more than 140 XML-based specifications for the travel industry.

The development of open specification messages based on XML, such as OTA schema, to ensure the interoperability between trading partners and working groups is not sufficiently expressive to guarantee an automatic exchange and processing of information to develop dynamic applications. A more appropriate solution is to use technologies from the semantic Web, such as ontologies, to deploy common language for tourism-related terminology and a mechanism for promoting the seamless exchange of information across all travel industry segments. Ontologies are the key elements enabling the shift from a purely syntactic to a semantic interoperability. An ontology can be defined as the explicit, formal descriptions of concepts and their relationships that exist in a certain universe of discourse, together with a shared vocabulary to refer to these concepts. With respect to an ontology a particular user group commits to, the semantics of data provided by the data sources to be integrated can be made explicit. Ontologies can be applied to the area of dynamic packaging to explicitly connect data and information from tourism information systems to its definition and context in machine-processable form.

Semantic digital libraries. Libraries are a key component of the information infrastructure indispensable for education. They provide an essential resource for students and researchers for reference and for research. Metadata has been used in libraries for centuries. For example, the two most common general classification systems, which use metadata, are the Dewey Decimal Classification (DDC) system and the Library of Congress Classification (LCC) system. The DDC system has 10 major subjects, each with 10 secondary subjects (DDC 2005). The LCC system uses letters instead of numbers to organize materials into 21 general branches of knowledge. The 21 subject categories are further divided into more specific subject areas by adding one or two additional letters and numbers (LCCS 2005).

As traditional libraries are increasingly converting to digital libraries, a new set of requirements has emerged. One important feature of digital libraries is the ability to efficiently browse electronic catalogues browsed. This requires the use of common metadata to describe the records of the catalogue (such as author, title, and publisher) and common controlled vocabularies to allow subject identifiers to be assigned to publications. The use of a common controlled vocabulary, thesauri, and taxonomy (Smrz, Sinopalnikova et al. 2003) allows search engines to ensure that the most relevant items of information are returned. Semantically annotating the contents of a digital library's database goes beyond the use of a controlled vocabulary, thesauri, or taxonomy. It allows retrieving books' records using meaningful information to the existing full text and bibliographic descriptions.

Semantic Web technologies, such as RDF and OWL, can be used as a common interchange format for catalogue metadata and shared vocabulary, which can be used by all libraries and search engines (Shum, Motta et al. 2000) across the Web. This is important since it is not uncommon to find library systems based on various metadata formats and built by different persons for their special purposes. By publishing ontologies, which can then be accessed by all users across the Web, library catalogues can use the same vocabularies for cataloguing, marking up items with the most relevant terms for the domain of interest. RDF and OWL provide a single and consistent encoding system so that implementers of digital library metadata systems will have their task simplified when interoperating with other digital library systems.

Semantic grid. The concept of Grid (Foster and Kesselman 1999) has been proposed as a fundamental computing infrastructure to support the vision of e-Science. The Grid is a service for sharing computer power and data storage capacity over the Internet and goes well beyond simple communication providing functionalities that enable the rapid assembly and disassembly of services into temporary groups.

Recently, the Grid has been evolving towards the Semantic Grid to yield an intelligent platform which allows process automation, knowledge sharing and reuse, and collaboration within a community (Roure, Jennings et al. 2001). The Semantic Grid is about the use of semantic Web technologies in Grid computing; it is an extension of the current Grid. The objective is to describe information, computing resources, and services in standard ways that can be processed by computers. Resources and services are represented using the technologies of the semantic Web, such as RDF. The use of semantics to locate data has important implications for integrating computing resources. It implies a two-step access to resources. In step one, a search of metadata catalogues is used to find the resources containing the data or service required by an application. In the second step, the data or service is accessed or invoked.

Semantic Web Search. Swoogle (Swoogle 2005) is a crawler-based indexing and retrieval system for the semantic Web built on top of the Google API. It was developed in the context of a research project of the Ebiquity research group at the Computer Science and Electrical Engineering Department of the University of Maryland, USA. In contrast to Google (Google 2005), Swoogle discovers, analyzes, and indexes Semantic Web Documents (SWD) written in RDF and OWL, rather than plain HTML documents. Documents are indexed using metadata about classes, properties, and individuals, as well as the relationships among them. Unlike traditional search engines, Swoogle aims to take advantage of the semantic metadata available in semantic Web documents. Metadata is extracted for each discovered document and relations (e.g. similarities) among documents are computed. Swoogle also defines an ontology ranking property for SWD which is similar to the pageRank (Brin and Page 1998) approach from Google and uses this information to sort search results. Swoogle provides query interfaces and services to Web users. It supports software agents, programs via service interfaces, and researchers working in the semantic Web area via the Web interface.

Semantic Bioinformatic Systems. The integration of information sources in the life sciences is one of the most challenging goals of bioinformatics (Kumar and Smith 2004). In this area, the Gene Ontology (GO) is one of the most significant accomplishments. The objective of GO is to supply a mechanism to guarantee the consistent descriptions of gene products in different databases. GO is rapidly acquiring the status of a *de facto* standard in the field of gene and gene product annotations (Kumar and Smith 2004). The GO effort includes the development of controlled vocabularies that describe gene products, establishing associations between the ontologies, the genes, and the gene products in the databases, and develop tools to create, maintain, and use ontologies (see http://www.geneontology.org/). GO has over 17,000 terms and it is organized in three hierarchies for molecular functions, cellular components, and biological processes (Bodenreider, Aubry et al. 2005).

Another well-known life science ontology is the Microarray Gene Expression Data (MGED) ontology. MGED provides standard terms in the form of an ontology organized into classes with properties for the annotation of microarray experiments (MGED 2005). These terms provide an unambiguous description of how experiments were performed and enable structured queries of elements of the experiments. The comparison between different experiments is only feasible if there is standardization in the terminology for describing experimental setup, mathematical post-processing of raw measurements, genes, tissues, and samples. The adoption of common standards by the research community for describing data makes it possible to develop systems for the management, storage, transfer, mining, and sharing of microarray data (Stoeckert, Causton et al. 2002). If data from every microarray experiment carried out by different research groups were stored with the same structure, in the same type of database, the manipulation of data would be relatively easy. Unfortunately, in practice, different research groups have very different requirements and, therefore, applications need mappings and translations between the different existing formats (Stoeckert, Causton et al. 2002).

7 Conclusions

Since its creation, the World Wide Web has allowed computers only to understand Web page layout for display purposes without having access to their intended meaning. The semantic Web aims to enrich the existing Web with a layer of machine-understandable metadata to enable the automatic processing of information by computer programs. The semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. To make possible the creation of the semantic Web the W3C (World Wide Web Consortium) has been actively working on the definition of open standards, such as the RDF (Resource Description Framework) and OWL (Web Ontology Language), and incentivate their use by both industry and academia. These standards are also important for the integration and interoperability for intra- and inter-business processes that have become widespread due to the development of business-to-business and business-to-customer infrastructures.

The semantic Web does not restrict itself to the formal semantic description of Web resources for machine-to-machine exchange and automated integration and processing. One important feature of formally describing resources is to allow computers to reason by inference. Once resources are described using facts, associations, and relationships, inference engines, also called reasoners, can derive new knowledge and draw logical conclusions from existing information. The use of inference engines in the semantic Web allows applications to inquire why a particular logical conclusion has been reached, i.e. semantic applications can give proof of their conclusions by explaining the steps involved in logical reasoning.

Even though the semantic Web is still in its infancy, there are already applications and tools that use this conceptual approach to build semantic Web based systems, ranging from the use of semantic Web services, semantic integration of tourism information sources, and semantic digital libraries to the development of bioinformatics ontologies.

8 References

Berners-Lee, T., J. Hendler, et al. (2001). The Semantic Web. Scientific American. **May 2001**.

Berners-Lee, T., J. Hendler, et al. (2001). The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American.

Bodenreider, O., M. Aubry, et al. (2005). Non-Lexical Approaches to Identifying Associative Relations in the Gene Ontology. Pacific Symposium on Biocomputing, Hawaii, USA, World Scientific.

BPEL4WS (2002). Web Services, IBM.

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual Web search engine. Seventh World Wide Web Conference, Brisbane, Australia.

Chinnici, R., M. Gudgin, et al. (2003). Web Services Description Language (WSDL) Version 1.2, W3C Working Draft 24, http://www.w3.org/TR/2003/WD-wsdl12-20030124/.

Christensen, E., F. Curbera, et al. (2001). W3C Web Services Description Language (WSDL).

Curbera, F., W. Nagy, et al. (2001). Web Services: Why and How. Workshop on Object-Oriented Web Services - OOPSLA 2001, Tampa, Florida, USA.

DAML (2001). DAML+OIL, http://www.daml.org/language/.
DC (2005). The Dublin Core Metadata Initiative, http://dublincore.org/.
DDC (2005). Dewey Decimal Classification, OCLC Online Computer Library Center, http://www.oclc.org/dewey/.
Decker, S., S. Melnik, et al. (2000). "The Semantic Web: The Roles of XML and RDF." Internet Computing **4**(5): 63-74.
DERI (2004). Digital Enterprise Research Institute (DERI).
DL (2005). Description Logics, http://www.dl.kr.org/.
FaCT (2005). FaCT++, http://owl.man.ac.uk/factplusplus/.
Fensel, D. (2001). Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Berlin, Springer-Verlag, http://www.cs.vu.nl/~dieter/ftp/paper/silverbullet.pdf.
Foster, I. and C. Kesselman (1999). The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann.
Gandon, F. L. and N. M. Sadeh (2003). OWL inference engine using XSLT and JESS, http://www-2.cs.cmu.edu/~sadeh/MyCampusMirror/OWLEngine.html.
Google (2005). Google Search Engine, www.google.com.
Grau, B. C. (2004). A Possible Simplification of the Semantic Web Architecture. WWW 2004, New York, USA.
Horrocks, I., F. v. Harmelen, et al. (2001). DAML+OIL, DAML.
Jasper, R. and M. Uschold (1999). A framework for understanding and classifying ontology applications. IJCAI99 Workshop on Ontologies and Problem-Solving Methods.
Jena (2005). Jena - A Semantic Web Framework for Java, http://jena.sourceforge.net/,.
Kumar, A. and B. Smith (2004). On Controlled Vocabularies in Bioinformatics: A Case Study in Gene Ontology. Drug Discovery Today: BIOSILICO. **2:** 246-252.
LCCS (2005). The Library of Congress, Library of Congress Classification System, http://www.loc.gov/catdir/cpso/lcco/lcco.html.
LSDIS (2004). METEOR-S: Semantic Web Services and Processes. **2004**.
MGED (2005). Microarray Gene Expression Data Society, http://www.mged.org/.
Mozilla (2005). The Mozilla project, http://www.mozilla.org/.
NISO (2005). Guidelines for the Construction, Format, and Management of Monolingual Thesauri. http://www.niso.org/standards/resources/z39-19a.pdf, National Information Standards Organization. **2005**.
OTA (2004). OpenTravel Alliance.
OWL (2004). OWL Web Ontology Language Reference, W3C Recommendation, World Wide Web Consortium, http://www.w3.org/TR/owl-ref/. **2004**.
OWL-S (2004). OWL-based Web Service Ontology. **2004**.
RDF (2002). Resource Description Framework (RDF), http://www.w3.org/RDF/.
RDFS (2004). RDF Vocabulary Description Language 1.0: RDF Schema, W3C, http://www.w3.org/TR/rdf-schema/.
Roure, D., N. Jennings, et al. (2001). Research Agenda for the Future Semantic Grid: A Future e-Science Infrastructure http://www.semanticgrid.org/v1.9/semgrid.pdf.
Sheth, A. (2003). Semantic Meta Data For Enterprise Information Integration. DM Review Magazine. **July 2003**.
Shum, S. B., E. Motta, et al. (2000). "ScholOnto: an ontology-based digital library server for research documents and discourse." International Journal on Digital Libraries **3**(3): 237-248.

Smrz, P., A. Sinopalnikova, et al. (2003). Thesauri and Ontologies for Digital Libraries. 5th Russian Conference on Digital Libraries (RCDL2003), St.-Petersburg, Russia.
SOAP (2002). Simple Object Access Protocol 1.1.
Stoeckert, C. J., H. C. Causton, et al. (2002). "Microarray databases: standards and ontologies." Nature Genetics **32**: 469 - 473.
Swoogle (2005). Search and Metadata for the Semantic Web - http://swoogle.umbc.edu/.
SWSI (2004). Semantic Web Services Initiative (SWSI). **2004**.
SWWS (2004). Semantic Web Enabled Web Service, Digital Enterprise Research Institute (DERI).
UDDI (2002). Universal Description, Discovery, and Integration.
Wielemaker, J. (2005). SWI-Prolog Semantic Web Library, http://www.swi-prolog.org/packages/semweb.html.
WSML (2004). Web Service Modeling Language (WSML). **2004**.
WSMO (2004). Web Services Modeling Ontology (WSMO). **2004**.
WSMX (2004). Web Services Execution Environment (WSMX). **2004**.
XML (2005). Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004, http://www.w3.org/TR/REC-xml/.
XMLSchema (2005). XML Schema, http://www.w3.org/XML/Schema.