Chapter IX

Metadata- and Ontology-Based Semantic Web Mining

Marie Aude Aufaure, Plateau du Moulon, France
Bénédicte Le Grand, Laboratoire d'Informatique de Paris 6, France
Michel Soto, Laboratoire d'Informatique de Paris 6, France
Nacera Bennacer, Plateau du Moulon, France

Abstract

The increasing volume of data available on the Web makes information retrieval a tedious and difficult task. The vision of the Semantic Web introduces the next generation of the Web by establishing a layer of machine-understandable data e.g. for software agents, sophisticated search engines and Web services. The success of the Semantic Web crucially depends on the easy creation, integration and use of semantic data. This chapter is a state-of-the-art review of techniques which could make the Web more “semantic”. Beyond this state-of-the-art, we describe open research areas and we present major current research programs in this domain.
This section presents the context and the challenges of semantic information retrieval. We also introduce the goals of Semantic Web (Berners-Lee et al., 2001) and data mining. Available data have become more and more complex; spatiotemporal parameters contribute to this complexity, as well as data’s lack of structure, multidimensionality, large volume, and dynamic evolution. Moreover, data formats and models are numerous, which makes their interoperability challenging. Biological databanks illustrate this situation. In the domain of tourism, queries can entail computations (e.g., in order to find the best path to a destination) including constraints which are not necessarily precisely formulated. Answers may be provided through the use of Web Services, and should be customized according to a user profile. Several Web Mining techniques have been proposed to enhance these different types of information retrieval, among which methods deriving from data analysis and from conceptual analysis. All these methods aim at making the Web more understandable but they differ in the way they deal with the complexity of data.

The increasing interest in Web information retrieval led to the Semantic Web initiative from the World Wide Web Consortium. The Semantic Web is not a new Web, but an extension of the existing one to make it more understandable to machines. The main goal is thus to express semantic information about data formally, so that this information may be processed and used by computers. Semantic information may appear as semantic annotations or metadata. Several formats have been designed to meet this goal, among which the Resource Description Framework (W3C, 1999) from the W3C and Topic Maps (ISO, 1999) from the International Standardisation Organisation. Both formats aim at describing resources and establish relationships among them. RDF can be enriched with a RDFS Schema which expresses class hierarchies and typing constraints, e.g. to specify that a given relation type can connect only specific classes. The semantic tagging provided by RDF and Topic Maps may be extended by references to external knowledge coming from controlled vocabularies, taxonomies, and ontologies. An ontology (Gruber, 1993) is an abstract model which represents a common and shared understanding of a domain. Ontologies generally consist of a list of interrelated terms and inference rules and can be exchanged between users and applications. They may be defined in a more or less formal way, from natural language to description logics. The Web Ontology Language (OWL) belongs to the latter category. OWL is built upon RDF and RDFS and extends them to express class properties.

Metadata and ontologies are complementary and constitute the Semantic Web’s building blocks. They avoid meaning ambiguities and provide more precise
answers. In addition to a better accuracy of query results, another goal of the Semantic Web is to describe the semantic relationships between these answers. The promises of the Semantic Web are numerous, but so are its challenges, starting with scalability. Semantic Web data are likely to increase significantly and associated techniques will have to evolve. The new tagging and ontology formats require new representation and navigation paradigms. The multiplicity of ontologies raises the issue of their integration; this area has been widely explored and solutions have been proposed, even though some problems still remain. The highly dynamic nature of the Semantic Web makes the evolution and maintenance of semantic tagging and ontologies difficult. The ultimate challenge is the automation of semantics extraction. This subject is developed in a whole section of this chapter. We study how traditional Web approaches might be used for a partial automation of knowledge extraction. Pages content and usage analysis are complementary to expand knowledge databases. However, this automation requires an evaluation of the extracted information.

This chapter is organized as follows: first, we introduce the notions of semantic metadata in general and ontologies in particular. Then we raise the issue of Semantic Web Mining (Berendt et al., 2002) and data integration, before studying how and to what extent the knowledge extraction process can be automated. We finally suggest some research directions for the future before concluding by presenting the limits of the Semantic Web’s extension.

---

**Metadata and Ontologies**

This section presents metadata representation formats, in particular RDF and Topic Maps, and their application to complex data. We also describe the concept of ontology and one associated standard, the Web Ontology Language (OWL). We study the added value of ontologies in comparison with simple metadata, in terms of expressivity and inference.

Let us first define metadata and annotations: metadata are data about data. An annotation is an explicative or critical note attached to a document, text or image. Web pages annotations become metadata when they are stored into a database or a server. We distinguish information attached to a resource from information stored and handled independently.

The Semantic Web can be divided into various layers of metadata, each level providing different degrees of expressivity, as shown in Figure 1 (Berners-Lee, 1998). In the following of this section, we describe Semantic Web formalisms, starting from the bottom of the stack.
XML, Namespaces and Controlled Vocabularies

XML is a first level of semantics which allows users to structure data with regard to their content rather than their presentation (Yergeau et al., 2004). XML tags may represent the meaning of data whereas HTML tags indicate the way data should be displayed.

Namespaces allow the unambiguous use of several vocabularies within a single document, by indicating explicitly which set a term belongs to. A controlled vocabulary is a set of terms defined by a community without giving any sense or organization among these terms. As an example, a book index is a controlled vocabulary. A very popular controlled vocabulary is the Dublin Core.

Dublin Core (www.dublincore.org) is a set of very simple elements used to describe various resources in terms of content (Title, Description, Subject, Source, Coverage, Type, Relationship), of intellectual property (Creator, Contributor, Editor, Rights) and of version (Date, Format, Identifier, Language). Dublin Core is composed of 15 elements which semantics have been established by an international consortium. This norm presents all the descriptive information found in traditional archive research systems, while preserving hierarchical relationships that exist between the different description levels. It facilitates the navigation into the hierarchical information structure.

Moreover, Dublin Core defines the categories of information that may be attached to a resource (Web page, document, or image) in order to enhance information retrieval. Dublin Core is used by a large community due to the following advantages:
• The set of elements is very simple, which makes this norm very easy to use for an efficient information retrieval;
• Its semantics is also easily understandable: Dublin Core helps beginner users find their way within data, by providing a common set of well defined and understood elements;
• Dublin Core is widely used; as an example, in 1999, it was translated into 20 languages;
• This norm is extensible; Dublin Core elements may be enriched with domain-specific information for particular communities.

RDF and Topic Maps

XML, controlled vocabularies and namespaces provide a first level of metadata. However, more semantics can be added with the Resource Description Framework (RDF) or Topic Maps standards. RDF was developed by the World Wide Web Consortium (W3C, 1999) whereas Topic Maps were defined by the International Organization for Standardization (ISO, 1999). Topic Maps do not appear on the Semantic Web stack shown on the figure 1, because there are not a W3C recommendation. On this figure, Topic Maps would be at the same level as RDF. The Topic Map paradigm was adapted to the Web by the TopicMaps.Org Consortium (TopicMaps.Org, 2001). Both RDF and Topic Maps aim at representing knowledge about information resources by annotating them. These paradigms are presented in the following subsections.

RDF

The Resource Description Framework (RDF) (W3C, 1999) syntax was designed to represent information about resources in the World Wide Web. Examples of such metadata are the author, creation and modification dates of a Web page. RDF provides a common framework for expressing semantic information about data so that it can be exchanged between applications without loss of meaning. RDF identifies things with Web identifiers (called Uniform Resource Identifiers, or URIs), and describes resources in terms of properties and property values.

Figure 2 shows the graphical RDF description of a Web page. This semantic annotation indicates that this page belongs to John Smith and that it was created on January 1, 1999 and modified on August 1, 2004. This corresponds to three RDF statements, giving information respectively on the author, creation and
Figure 2. Example RDF graph

As shown in Figure 2, statements about resources can be represented as a graph of nodes and arcs corresponding to the resources, their properties and their values. RDF provides an XML syntax (called serialisation syntax) for these graphs. The following code is the XML translation of the graph in Figure 2:

modification dates of this page. Each statement consists of a (Resource, Property, Value) triplet. In our example:

- http://www.foo.com/~smith is a resource
- The element <author> is a property
- The string « John Smith » is a value.

A statement may also be described in terms of (Subject, Predicate, Object):

- The resource http://www.foo.com/~smith is the subject
- The property <author> is the predicate
- The value « John Smith » is the object.

As shown in Figure 2, statements about resources can be represented as a graph of nodes and arcs corresponding to the resources, their properties and their values. RDF provides an XML syntax (called serialisation syntax) for these graphs. The following code is the XML translation of the graph in Figure 2:
Topic Maps (ISO, 1999) are an ISO standard which describes knowledge and links it to existing information resources. RDF and Topic Maps thus have similar goals.

Although Topic Maps allow organizing and representing very complex structures, the basic concepts of this model (topics, occurrences, and associations) are simple. A topic is a syntactic construct which corresponds to the expression of a real-world concept in a computer system. The figure 3 represents a very small Topic Map which contains four topics: EGC 2005, Paris, Ile-de-France and France. These topics are instances of other topics: EGC 2005 is a conference, Paris is a city, Ile-de-France is a region and France is a country. A topic type is a topic itself, which means that conference, city, region and country are also topics.

A topic may be linked to several information resources (e.g., Web pages) which are considered to be somehow related to this topic. These resources are called occurrences of a topic. Occurrences provide means of linking real resources to abstract concepts, which helps organize data and understand their context.

An association adds semantics to data by expressing a relationship between several topics, such as EGC 2005 takes place in Paris, Paris is located in Ile-de-France, and so on. Every topic involved in an association plays a specific role in this association, for example, Ile de France plays the role of container and Paris plays the role of containee.

It is interesting to notice that topics and information resources belong to two different layers. Users may navigate at an abstract level (the topic level) instead of navigating directly within data.

RDF and Topic Maps both add semantics to existing data without modifying them. They are two compatible formalisms: Moore (2001) stated that RDF could be used to model Topic Maps and vice versa. There are slight differences, for
example, the notion of *scope* (context) exists in Topic Maps and not in RDF. RDF is more synthetic and better adapted to queries whereas Topic Maps are better for navigation purposes.

So far, we have described the lower layers of the Semantic Web stack; in the next section, we will describe more expressive formalisms: ontologies. We will also describe two other formalisms, which are not specific to the Web: taxonomies and thesauri.

**Taxonomies, Thesauri and Ontologies**

**Taxonomies and Thesauri**

Taxonomies and thesauri do not appear on the Semantic Web stack as they were not specifically designed for the Web; they, however, belong to the Semantic Web picture. In this section, we define these notions and we indicate their level in the stack.

A *taxonomy* is a hierarchically-organized controlled vocabulary. The world has many taxonomies, because human beings naturally classify things. Taxonomies
are semantically weak. According to Daconta et al. (2003), taxonomies are commonly used when navigating without a precise research goal in mind.

A thesaurus is a “controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardized relationship indicators” (ANSI/NISO Z39.19-1993 (R1998), p.1.). The purpose of a thesaurus is to facilitate documents retrieval. The WordNet thesaurus (Miller, 1995) organizes English nouns, verbs, adverbs, and adjectives into a set of synonyms and defines relationships between synonyms.

Both taxonomies and thesauri provide a vocabulary of terms and simple relationships between these terms. Therefore, taxonomies and thesauri are above XML, namespaces and controlled vocabulary in the Semantic Web stack. However, the relationships they express are not as rich as the ones provided by RDF or Topic Maps and consequently by ontologies.

**Ontologies**

**Definitions**

As we saw earlier, Berners-Lee (1998) proposed a layered architecture for the Semantic Web languages, among which were XML, XMLSchema, RDF, and RDFS Schema (RDFS). RDFS defines classes and properties (binary relation), range and domain constraints on properties, subclass and subproperty as subsumption relations. However, RDFS is insufficient in terms of expressivity; this is also true for Topic Maps. On the other hand, ontologies allow a better specification of constraints on classes. They also make reasoning possible, as new knowledge may be inferred, e.g., by transitivity. **Ontologies** aim at formalizing domain knowledge in a generic way and provide a common agreed understanding of a domain, which may be used and shared by applications and groups.

In computer science, the word ontology, borrowed from philosophy, represents a set of precisely defined terms (vocabulary) about a specific domain and accepted by this domain’s community. An ontology thus enables people to agree upon the meaning of terms used in a precise domain, knowing that several terms may represent the same concept (synonyms) and several concepts may be described by the same term (ambiguity). Ontologies consist in a hierarchical description of important concepts of a domain, and in a description of each concept’s properties. Ontologies (Gomez-Perez et al., 2003) are at the heart of information retrieval from nomadic objects, from the Internet and from heterogeneous data sources.

Ontologies generally consist of a taxonomy — or vocabulary — and of inference rules such as transitivity and symmetry. They may be used in conjunction with
RDF or Topic Maps e.g. to allow consistency checking or to infer new information.

According to Gruber (1993), “an ontology is an explicit specification of a conceptualization.”

Jeff Heflin, editor of the OWL Use Cases and Requirements (Heflin, 2004), considers that “an ontology defines the terms used to describe and represent an area of knowledge. [...] Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them. [...] Ontologies are usually expressed in a logic-based language, so that detailed, accurate, consistent, sound, and meaningful distinctions can be made among the classes, properties, and relations.”

Berners-Lee et al. (2001) say that “Artificial-intelligence and Web researchers have co-opted the term for their own jargon, and for them an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules.”

Ontologies may be classified as follows:

Guarino (1998) classifies ontologies according to their level of dependence with regard to a specific task or point of view. He distinguishes four categories: high-level, domain, task, and application ontologies.

Lassila and McGuinness (2001) categorize ontologies according to their expressiveness and to the richness of represented information. Depending on the domain and the application, an ontology may be more or less rich, from a simple vocabulary to real knowledge bases; it may be a glossary where each term is associated to its meaning in natural language. It may also be a thesaurus in which terms are connected through semantic links (synonyms in WordNet) or even genuine knowledge bases comprising notions of concepts, properties, hierarchical links, and properties constraints.

After defining the concept of ontology, we now present ontology languages.

**Ontology Languages**

The key role that ontologies are likely to play in the future of the Web has led to the extension of Web markup languages. In the context of the Semantic Web, an ontology language should:
• be compatible with existing Web standards,
• define terms precisely and formally with adequate expressive power,
• be easy to understand and use,
• provide automated reasoning support,
• provide richer service descriptions which could be interpreted by intelligent agents,
• be sharable across applications.

Ontology languages can be more or less formal. The advantage of formal languages is the reasoning mechanisms which appear in every phase of conception (satisfiability, subsumption, etc.), use (query, instantiation) and maintenance of an ontology (consistency checking after an evolution). The complexity of underlying algorithms depends on the power and the semantic richness of the used logics.

When querying an ontology, a user does generally not have the global knowledge of the ontology schema. The language should thus allow him to query both the ontology schema and its instances in a consistent manner. The use of description logics (DL), a subset of first-order logic, unifies the description and the manipulation of data. In DL, the knowledge base consists of a T-Box (Terminological-Box) and of a A-Box (Assertional-Box). The T-Box defines concepts and relationships between concepts, whereas the A-Box consists of assertions describing a situation (Nakabasami, 2002).

At the description level, concepts and roles are defined; at the manipulation level, the query is seen as a concept and reasoning mechanisms may be applied. For instance, the description of a query may be compared to an inconsistent description. If they are equivalent, this means that the user made a mistake in the formulation of his query (remind that he does not know the ontology schema). The query may also be compared (by subsumption) to the hierarchy of concepts (the ontology). One limit of description logics is that queries can only return existing objects, instead of creating new objects, as database query languages such as SQL can do.

In the next section, we focus on a specific ontology language: the Web Ontology Language (OWL).

**OWL**

To go beyond the “plain text” searching approach it is necessary to specify the semantics of the Web resources content in a way that can be interpreted by
intelligent agents. The W3C has designed the Web Ontology Language: OWL (W3C, 2004) (Dean & Schreiber, 2003), a semantic markup language for Web resources, as a revision of the DAML+OIL (Horrocks, 2002). It is built on W3C standards XML, RDF/RDFS (Brickley & Guha, 2003; Lassila & Swick, 1999) and extends these languages with richer modeling primitives. Moreover, OWL is based on description logics (Baader et al., 2003; Horrocks & Patel-Schneider, 2003; Horrocks et al., 2003); OWL may then use formal foundations of description logic, mainly known reasoning algorithms and implemented systems (Volker & Möller, 2001; Horrocks, 1998).

OWL allows:

• the formalization of a domain by defining classes and properties of those classes,
• the definition of individuals and the assertion of properties about them, and
• the reasoning about these classes and individuals.

We saw in the previous section that RDF and Topic Maps lacked expressive power; OWL, layered on top of RDFS, extends RDFS’s capabilities. It adds various constructors for building complex class expressions, cardinality restrictions on properties, characteristics of properties, and mapping between classes and individuals (W3C, 2004) (Dean & Schreiber, 2003). An ontology in OWL is a set of axioms describing classes, properties, and facts about individuals.

The following basic example of OWL illustrates these concepts:

In this example, Man and Woman are defined as subclasses of the Person class; hasParent is a property that links two persons. hasFather is a subproperty of hasParent and its range is constrained to the Man Class. hasChild is the inverse property of hasParent.

```xml
<owl:Class rdf:ID="Person"/>

<owl:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>

<owl:Class rdf:ID="Woman">
  <rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
```

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
Although OWL is more expressive than RDFS or Topic Maps, it still has limitations; in particular, it lacks a more powerful language to better describe properties, in order to provide more inference capabilities. An extension to OWL with Horn-style rules has been proposed by Horrocks and Patel Schneider (2004), called ORL: OWL Rules Language. ORL itself may be further extended if more expressive power is needed.

Semantic Web Information Retrieval

Semantic Web Mining aims at integrating the areas of Semantic Web and Web Mining (Berendt et al., 2002). The purpose is twofold:

- improve Web mining efficiency by using semantic structures such as ontologies, metadata, thesauri, and
- use Web mining techniques and learn ontologies from Web resources as automatically as possible and thus help building the Semantic Web.
We present the benefits of metadata and ontologies for a more relevant information retrieval, as shown on figure 4 (Decker et al., 2000). The use of controlled vocabularies avoids meaning conflicts, whereas ontologies allow semantic data integration. Results can be customized through the use of semantic annotations.

Figure 4 shows the various components of a semantic information retrieval from Web pages. Automated agents use various Semantic Web mechanisms in order to provide relevant information to end users or communities of users. To achieve this goal, Web pages must be annotated, using the terms defined in an ontology (Ontology Construction Tool). Once the pages are semantically annotated, agents use existing metadata and inference engines to answer queries. If a query is formulated with a different ontology, a semantic integration is performed with the Ontology Articulation Toolkit.
Information Retrieval in the Semantic Web

In this section, we show how semantic metadata enhance information retrieval. References to ontologies avoid ambiguities and therefore allow advanced queries and provide more relevant answers to precise information needs. We define a search as precise if the information need can be formally specified with a query language. However, it is not always possible to formulate a precise query, for example if what is looked for is an overview of a set of Web pages. Typically, this is the case when one follows HTTP hyperlinks during a Web navigation. In order to meet the goals of these fuzzy searches, the semantic relationships defined by RDF graphs or Topic Maps are very helpful, as they connect related concepts. Thus, Semantic Web techniques are complementary and they benefit both precise and fuzzy searches.

An implementation of information retrieval prototypes based on RDF and Topic Maps was achieved in the OmniPaper project (Paepen et al., 2002), in the area of electronic news publishing. In both cases, the user submits a natural-language query to a large set of digital newspapers. Searches are based on linked keywords which form a navigation layer. User evaluation showed that the semantic relations between articles were considered very useful and important for relevant content retrieval.

Semantic Web methodologies and tools have also been implemented in an IST/CRAFT European Program called Hi-Touch, in the domain of tourism (Euzénat et al., 2003). In the Hi-Touch platform, Semantic Web Technologies are used to store and organize information about customers’ expectations and tourism products. This knowledge can be processed by machines as well as by humans in order to find the best matching between supply and demand. The knowledge base system combines RDF, Topic Maps, and ontologies.

The figure 5 illustrates a semantic query performed with Mondeca’s Intelligent Topic Manager (http://www.mondeca.com), in the context of the Hi-Touch project. Users can express their queries with keywords, but they can also specify the type of result they expect, or provide more details about its relationships with other concepts. Figure 5 also shows the graphical environment, centered on the query result, which allows users to see its context.

Semantic Integration of Data

The Web is facing the problem of accessing a dramatically increasing volume of information generated independently by individual groups, working in various domains of activity with their own semantics. The integration of these various semantics is necessary in the context of the Semantic Web because it allows the
capitalization of existing semantic repositories such as ontologies, taxonomies, and thesauri. This capitalization is essential for reducing cost and time on the Semantic Web pathway.

A semantic integration allows to share data that exhibits a high degree of semantic heterogeneity, notably when related or overlapping data encompass different levels of abstraction, terminologies or representations. Data available in current information systems is heterogeneous both in its content and its representation formalism. Two common data integration methods are mediators and data warehouses.

The warehouse approach provides a global view, by centralizing relevant data. Access to data is fast and easy and thus data warehouses are useful when complex queries and analyses are needed. However, the limits of this approach are the required storage capability and the maintenance of the warehouse content. With this approach updates may be performed using different techniques:
periodical full reconstruction: this is the most commonly used and simplest technique, but this is also the most time-consuming method,

periodical update: incremental approach for updating the data warehouse with the difficulty of detecting the changes within the multiple sources of data.

immediate update: another incremental approach which aims at keeping the data as consistent as possible. This technique may consume a lot of communication resources; thus it can only be used for small data warehouses built on data sources with a low update rate.

On the other hand, the mediator approach keeps the initial distribution of data. The mediator can be seen as an interface between users and data sources during a query. The data mediator architecture provides a transparent access to heterogeneous and distributed data sources and eliminates the problem of data update (Ullman, 1997). Initial queries are expressed by users with the global schema provided by the mediator and reformulated in sub queries on the data sources. Answers are then collected and merged according to the global schema (Halevy, 2001).

There are currently two approaches for building a global schema for a mediator: global as view (GAV) and local as view (LAV). With the first approach, a global schema is built using the terms and the semantics of data sources. As a consequence, query reformulation is simple but the addition of a new data source modifies the global schema. Thus, the global as view approach does not scale very well. With the second approach, the global schema is built independently from the data sources. Each data source is defined as a view on the global schema using the terms and the semantic of the global schema. Thus, adding or removing a new data source is easy but query reformulation is much more complex. Nevertheless, the local as view approach is currently preferred for its scaling capabilities. Consequently, a lot of work is done on query reformulation where ontologies play a central role as they help to express queries. A third approach named GLAV aims at combining advantages of GAV and LAV by associating views over the global schema to views over the data sources (Cali, 2003). Both GAV and LAV approaches consider data sources as sets of relations from data bases. This appears to be inadequate in the context of Web data integration because of the necessary navigation through hyperlinks to obtain the data. Combining the expressivity of GAV and LAV allows to formulate query execution plans which both query and navigate the Web data sources (Friedman et al., 1999).
Ontologies Integration and Evolution

The success of the Semantic Web depends on the expansion of ontologies. While many people and organizations develop and use knowledge and information systems, it seems obvious that they will not use a common ontology. As ontologies will proliferate, they will also diverge; many personalized and small-scale conceptualizations will appear. Accessing the information available on the Semantic Web will be possible only if these multiple ontologies are reconciled.

Ontology integration and evolution should take advantage of the work already done in the database field for schema integration and evolution (Rahm & Bernstein, 2001; Parent & Spaccapietra, 1998). Automatic schema matching led to a lot of contributions in schema translation and integration, knowledge representation, machine learning, and information retrieval. Schema integration and ontology integration are quite similar problems. Database schemas can be well-structured (relational databases) or semistructured (XML Schemas). Integrity constraints and cardinality are of great importance in these structures. However, ontologies are semantically richer than database schemas; they may also integrate rules and be defined formally (using description logics). Database schema integration takes instances into account while instances are less important in the case of ontologies (we do not always have instances for a given ontology).

Schema integration is studied since the beginning of the 1980s. The goal is to obtain a global view of a set of schemas developed independently. The problem is that structures and terminologies are different because these schemas have been designed by different persons. The approach consists in finding relationships between different schemas (matching), and then in unifying the set of correspondences into an integrated and consistent schema.

Mechanisms for ontologies integration aim at providing a common semantic layer in order to allow applications to exchange information in semantically sound manners. Ontologies integration has been the focus of a variety of works originating from diverse communities entailing a large number of fields from machine learning and formal theories to heuristics, database schema and linguistics. Relevant terms encountered in these works include merging, alignment, integration, mapping, and matching. Ontology merging aims at creating a new ontology from several ontologies. The objective is to build a consistent ontology containing all the information from the different sources. Ontology alignment makes several ontologies consistent through a mutual agreement. Ontology integration creates a new ontology containing only parts of the source ontologies. Ontology mapping defines equivalence relations between similar concepts or relations from different ontologies. Ontology matching (Doan et al., 2003) aims at finding the semantic mappings between two given ontologies.
(Hammed et al., 2004) review several architectures for multiple-ontology systems at a large scale. The first architecture is “bottom-up” and consists in mappings between pairs of ontologies. In this case, the reconciliation is done only when necessary and not for all ontologies. The advantages of such an approach are its simplicity (because of the absence of a common ontology) and its flexibility (the mappings are performed only if necessary and can be done by the designers of the individual ontologies). The main drawback comes from the number of mappings to do when many ontologies are taken into account. Another drawback is that there is no attempt to find common conceptualizations.

The second approach maps the ontologies towards a common ontology. In this case, mapping an ontology O1 to another ontology O2 consists firstly in mapping O1 to the common ontology and secondly in mapping from the common ontology to O2. The advantage is that it reduces the number of mappings and the drawback is the development cost of the common ontology which has to be sufficiently expressive to allow mappings from the individual ontologies. An alternative approach consists in building clusters of common ontologies and in defining mappings between these clusters. In this case, individual ontologies map with one common ontology and mappings between the common ontologies are also defined. This approach reduces the number of mappings and finds common conceptualizations, which seems more realistic in the context of the Semantic Web.

Several tools have been developed to provide support for the construction of semantic mappings. Underlying approaches are usually based on heuristics that identify structural and naming similarities. They can be categorized according to the type of inputs required for the analysis: descriptions of concepts in OBSERVER (Mena & al., 2000), concept hierarchies in iPrompt and AnchorPrompt (Noy et al., 2003) and instances of classes in GLUE (Doan and al., 2003) and FCA-Merge (Stumme & Maedche, 2001). The automated support provided by these tools significantly reduces the effort required by the user. Approaches designed for mapping discovery are based upon machine learning techniques and compute similarity measures to extract mappings. In this section, we present the FCA-Merge method (Stumme & al., 2001) for ontology merging, the GLUE system (Doan et al., 2003) based on a machine learning approach and the iPrompt method.

FCA-Merge (Stumme & Maedche, 2001) is based on formal concept analysis and lattice generation and exploration. The input of the method is a set of documents, representative of a particular domain, from which concepts and the ontologies to merge are extracted. This method is based on the strong assumption that the documents cover all concepts from both ontologies. The concept lattice is then generated and pruned. Then, the construction of the merged ontology is semi-automatic.
GLUE (Doan et al., 2003) employs machine learning techniques to find mappings between two ontologies; for each concept from one ontology, GLUE finds the most similar concept in the other ontology using probabilistic definitions of several similarity measures. The similarity measure between two concepts is based on conditional probabilities. A similarity matrix is then generated and GLUE uses some common knowledge and domain constraints to extract the mappings between two ontologies. That knowledge includes domain-independent knowledge such as “two nodes match if nodes in their neighbourhood also match” as well as domain-dependant knowledge such as “if node Y is a descendant of node X, and Y matches professor, then it is unlikely that X matches assistant professor”. GLUE uses a multilearning strategy and exploits the different types of information a learner can obtain from the training instances and the taxonomic structure of ontologies.

The iPrompt method (Noy et al., 2003) is dedicated to ontology merging; it is defined as a plug-in on Protégé2000 (Noy et al., 2001). The semi-automatic algorithm is the following:

- make initial suggestions for merging (executed manually by the user),
- select an operation (done by the user according to a particular focus),
- perform automatic updates,
- find conflicts,
- update the initial list of suggestions.

Other approaches focus on the specification and formalization of inter-schema correspondences. (Calvanese et al. 2001) propose a formal framework for Ontology Integration Systems. Ontologies in their framework are expressed as Description Logic (DL) knowledge bases, and mappings between ontologies are expressed through suitable mechanisms based on queries. Two approaches are proposed to realize this query/view-based mapping: global-centric and local-centric. In the global-centric approach, the mapping is specified by associating to each relation in the global schema one relational query over source relations; on the other hand, the local-centric approach relies on reformulation of the query in terms of the queries to the local sources.

Ontology evolution (Noy et al., 2004) is rather similar to ontology merging; the difference relies in finding differences rather than similarities between ontologies. Ontology evolution and versioning should also benefit from the work done in the database community. Ontologies change over time. (Noy et al., 2004) describe changes that can occur in an ontology: changes in the domain (comparable with database schema evolution), changes in conceptualization (application
or usage points of view), and changes in the explicit specification (transformation from a knowledge representation language to another). The compatibility between different versions is defined as follows: instance-data preservation, ontology preservation (a query result obtained with the new version is a superset of those obtained with the old version), consequence preservation (in the case of an ontology treated as a set of axioms, the inferred facts from the old version can also be inferred with the new version), and consistency preservation (the new version of the ontology does not introduce logical inconsistencies). An open research issue in this field is the development of algorithms for automatically finding differences between versions.

In this section we explained how the Semantic Web will enhance information retrieval and data mining. However, we have seen that the success of the Semantic Web required the integration of data and ontologies. Another (obvious) prerequisite is the existence of semantic metadata. The next section presents current techniques and open research areas in the domain of automatic extraction of semantic metadata.

**Automatic Semantics Extraction**

Information retrieval provides answers to precise queries, whereas Data Mining brings an additional view for the understanding and the interpretation of data, which can be materialized with metadata. This section is more prospective and tackles current work in the field of the extraction of concepts, relationships between concepts, and metadata. We show how ontologies may enhance knowledge extraction through data mining methods. This will allow a partial automation of semantic tagging and will ease the update and maintenance of metadata and ontologies. Evaluation methods will have to be defined in order to check the validity of extracted knowledge.

**Tools and Methods For Manual Ontology Construction**

Most existing ontologies have been built manually. The first methodologies we can find in the literature (Ushold & King, 1995; Grüninger & Fox, 1995) have been defined taking into account enterprise ontologies development.

Based on the experience of the Tove project, Grüninger and Fox’s methodology is inspired by the development of knowledge-based systems using first order logic. They first identify the main scenarios and they elaborate a set of informal competency questions that the ontology should be able to answer. The set of
questions and answers are used to extract the main concepts and their relationships and properties which are formalized using first-order logic. Finally, we must define the conditions under which the solutions of the questions are complete. This methodology provides a basis for ontology construction and validation. Nevertheless, some support activities such as integration and acquisition are missing, as well as management functions (e.g., planification, quality control).

*Methontology* (Gomez-Pérez et al., 2003) builds ontologies from scratch; this methodology also enables ontology re-engineering (Gomes-Perez & Rojas, 1999). Ontological re-engineering consists in retrieving a conceptual model from an ontology, and transforming it in a more suitable one. *Methontology* enables the construction of ontologies at the “knowledge level”. This methodology consists in identifying the ontology development process with the following main activities: evaluation, configuration, management, conceptualization, integration, and implementation. A lifecycle is based on evolving prototypes. The methodology specifies the steps to perform each activity, the techniques used, the products to be output and how they are to be evaluated. This methodology is partially supported by *WebODE* and many ontologies have been developed in different fields.

The *DOGMA* modeling approach (Jarrar & Meersman, 2002) comes from the database field. Starting from the statement that integrity constraints may vary from one application to another and that the schema is more constant, they propose to split the ontology in two parts. The first one holds the data structure and is application-independent, and the second one is a set of commitments dedicated to one application.

*On-To-Knowledge* is a process-oriented methodology for introducing and maintaining ontology-based knowledge management systems (Staab et al., 2001); it is supported by the *OntoEdit* Tool. *On-To-Knowledge* has a set of techniques, methods, and principles for each of its processes (feasibility study, ontology kickoff, refinement, evaluation, and maintenance) and indicates the relationships between the processes. This methodology takes usage scenarios into account and is consequently highly application-dependant.

Many tools and methodologies exist for the construction of ontologies. Their differences are the expressiveness of the knowledge model, the existence of an inference and query engine, the type of storage, the formalism generated and its compatibility with other formalisms, the degree of automation, consistency checking, etc.

These tools may be divided into two groups:

- Tools for which the knowledge model is directly formalized in an ontology language:
\[\text{o} \quad \text{Ontolingua Server (Ontolingua et KIF)},\]
\[\text{o} \quad \text{OntoSaurus (Loom)},\]
\[\text{o} \quad \text{OLied (OIL then DAML+OIL then OWL) DL, consistency checking and classification using inference engines such as Fact and Racer.}\]

- Tools for which the knowledge model is independent from the ontology language:
  \[\text{o} \quad \text{Protégé-2000},\]
  \[\text{o} \quad \text{WebODE},\]
  \[\text{o} \quad \text{OntoEdit},\]
  \[\text{o} \quad \text{KAON}.\]

The most frequently cited tools for ontology management are \textit{OntoEdit}, \textit{Protégé-2000} and \textit{WebODE}. They are appreciated for their n-tiers architecture, their underlying database support, their support of multilingual ontologies and for their methodologies of ontology construction.

In order to reduce the effort to build ontologies, several approaches for the partial automation of the knowledge acquisition process have been proposed. They use natural language analysis and machine learning techniques.

**Concepts and Relationships Extraction**

Ontology learning (Maedche, 2002) can be seen as a plug-in in the ontology development process. It is important to define which phases may be automated efficiently. Appropriate data for this automation should also be defined. Existing ontologies should be reused using fusion and alignment methods. A priori knowledge may also be used. One solution is to provide a set of algorithms to solve a problem and combine results. An important issue about ontologies is their adaptation to different domains, as well as their extension and evolution.

When data is modeled with schemas, the work achieved during the modeling phase can be used for ontology learning. If a database schema exists, existing structures may be combined into more complex ones, and they may be integrated through semantic mappings. If data is based on Web schemas, such as DTDs or XML schemas, ontologies may be derived from these structures. If data is defined with instances, ontology learning may be done with conceptual clustering and A-Box mining (Nakabasami, 2002). With semistructured data, the goal is to find the implicit structure.
The most common type of data used for ontology learning is natural language data, as can be found in Web pages. In recent years, research aimed at paving the way and different methods have been proposed in the literature to address the problem of (semi) automatically deriving a concept hierarchy from text. Much work in a number of disciplines (computational linguistics, information retrieval, machine learning, databases, software engineering) has actually investigated and proposed techniques for solving part of the overall problem.

The notion of ontology learning is introduced as an approach that may facilitate the construction of ontologies by ontology engineers. It comprises complementary disciplines that feed on different types of unstructured and semistructured data in order to support a semiautomatic, cooperative ontology engineering process characterized by a coordinated interaction with human modelers.

Resource processing consists in generating a set of pre-processed data as input for the set of unsupervised clustering methods for automatic taxonomy construction. The texts are preprocessed, enriched by background knowledge using stopword, stemming, and pruning techniques. Strategies for disambiguation by context are applied.

Clustering methods organize objects into groups whose members are similar in some way. These methods operate on vector-based semantic representations which describe the meaning of a word of interest in terms of counts of its co-occurrence with context words appearing within some delineation around the target word. The use of a similarity/distance measure in order to compute the similarity/distance between vectors of terms in order to decide if they are semantically similar and thus should be clustered or not.

In general, counting frequencies of terms in a given set of linguistically preprocessed documents of a corpus is a simple technique that allows extracting relevant lexical entries that may indicate domain concepts. The underlying assumption is that a frequent term in a set of domain-specific texts indicates the occurrence of a relevant concept. The relevance of terms is measured according to the information retrieval measure tfidf (term frequency inverted document frequency).

More elaborated approaches are based on the assumption that terms are similar because they share similar linguistic contexts and thus give rise to various methods which group terms based on their linguistic context and syntactic dependencies.

We now present related work in the field of ontology learning.

Faure and Nedellec (1998) present an approach called ASIUM, based on an iterative agglomerative clustering of nouns appearing in similar contexts. The user has to validate the clusters built at each iteration. ASIUM method is based on conceptual clustering; the number of relevant clusters produced is a function of the percentage of the corpus used.
In Cimiano et al. (2004) the linguistic context of a term is defined by the syntactic dependencies that it establishes as the head of a subject, of an object or of a PP-complement with a verb. A term is then represented by its context using a vector, the entries of which count the frequency of syntactically dominating verbs.

Pereira et al. (1993) present a divisive clustering approach to build a hierarchy of nouns. They make use of verb-object relations to represent the context of a noun. The results are evaluated by considering the entropy of the produced clusters and also in the context of a linguistic decision task.

Caraballo (1999) uses an agglomerative technique to derive an unlabeled hierarchy of nouns through conjunctions of nouns and appositive constructs. The approach is evaluated by presenting the hypernyms and the hyponym candidates to users for validation.

Bisson et al. (2000) present a framework and its corresponding workbench (Mo’K) that supports the development of conceptual clustering methods to assist users in an ontology building task. It provides facilities for evaluation, comparison, characterization of different representations, as well as pruning parameters and distance measures of different clustering methods.

Most approaches have focused only on discovering taxonomic relations, although nontaxonomic relations between concepts constitute a major building block in common ontology definitions. In Maedche et al. (2000) a new approach is described to retrieve nontaxonomic conceptual relations from linguistically processed texts using a generalized association rule algorithm. This approach detects relations between concepts and determines the appropriate level of abstraction for those relations. The underlying idea is that frequent couplings of concepts in sentences can be regarded as relevant relations between concepts. Two measures evaluate the statistical data derived by the algorithm: \textit{Support} measures the quota of a specific coupling within the total number of couplings. \textit{Confidence} denotes the part of all couplings supporting both domain and range concepts within the number of couplings that support the same domain concept. The retrieved measures are propagated to super concepts using the background knowledge provided by the taxonomy. This strategy is used to emphasize the couplings in higher levels of the taxonomy. The retrieved suggestions are presented to the user. Manual work is still needed to select and name the relations.

Verbs play a critical role in human languages. They constrain and interrelate the entities mentioned in sentences. The goal in Wiemer-Hastings et al. (1998) is to find out how to acquire the meanings of verbs from context.

In this section, we focused on the automation of semantics extraction. The success of such initiatives is crucial to the success of the Semantic Web, as the volume of data does not allow a completely manual annotation. This subject remains an open research area.
In the next section, we present other research areas which we consider as strategic for the Semantic Web.

Future Trends

Web Content and Web Usage Mining Combination

One interesting research topic is the exploitation of users profiles and behaviour models in the data mining process, in order to provide personalized answers. The Web mining (Kosala & Blockeel, 2000) is a data mining process applied to the Web. Vast quantities of information are available on the Web and Web mining has to cope with its lack of structure. Web mining can extract patterns from data trough content mining, structure mining and usage mining. Content mining is a form of text mining applied to Web pages. This process allows to discover relationships related to a particular domain, co-occurrences of terms in a text, and so forth. Knowledge is extracted from a Web page. Structure mining is used to examine data related to the structure of a Web site. This process operates on Web pages’ hyperlinks. Structure mining can be considered as a specialisation of Web content mining. Web usage mining is applied to usage information such as logs files. A log file contains information related to the queries executed by users to a particular Web site. Web usage mining can be used to modify the Web site structure or give some recommendations to the visitor. Personalisation can also be enhanced by usage analysis.

Web mining can be useful to add semantic annotations (ontologies) to Web documents and to populate these ontological structures. As stated below, Web content and usage mining should be combined to extract ontologies and to adapt them to the usage.

Ontology creation and evolution require the extraction of knowledge from heterogeneous sources. In the case of the Semantic Web, the knowledge extraction is done from the content of a set of Web pages dedicated to a particular domain. Web pages are semistructured information. Web usage mining extracts navigation patterns from Web log files and can also extract information about the Web site structure and user profiles. Among Web usage mining applications, we can point out personalization, modification and improvement of Web site, detailed description of a Web site usage. The combination of Web content and usage mining could allow to build ontologies according to Web pages content and refine them with behaviour patterns extracted from log files. Web usage mining provides more relevant information to users and it is therefore a very powerful tool for information retrieval. Another way to provide more
accurate results is to involve users in the mining process, which is the goal of visual data mining, described in the next section.

Visualization

Topic Maps, RDF graphs, and ontologies are very powerful but they may be complex. Intuitive visual user interfaces may significantly reduce the cognitive load of users when working with these complex structures. Visualization is a promising technique for both enhancing users’ perception of structure in large information spaces and providing navigation facilities. According to Gershon and Eick (1995), it also enables people to use a natural tool of observation and processing (their eyes as well as their brain) to extract knowledge more efficiently and find insights.

The goal of semantic graphs visualization is to help users locate relevant information quickly and explore the structure easily. Thus, there are two kinds of requirements for semantic graphs visualization: representation and navigation. A good representation helps users identify interesting spots whereas an efficient navigation is essential to access information rapidly. We both need to understand the structure of metadata and to locate relevant information easily.

A study of representation and navigation metaphors for Semantic Web visualization has been studied by Le Grand and Soto (to appear). Figure 6 shows two example metaphors for the Semantic Web visualization: a 3D cone-tree and a virtual city. In both cases, the semantic relationships between concepts appear on the display, graphically, or textually.

Many open research issues remain in the domain of Semantic Web visualization; in particular, evaluation criteria must be defined in order to compare the various existing approaches. Moreover, scalability must be addressed, as most current visualization tools can only represent a limited volume of data.

Semantic Web services are also an open research area and are presented in the next section.

Semantic Web Services

Web services belong to the broader domain of service-oriented computing (Papazoglou, 2003) where the application development paradigm relies on a loosely coupling of services. A service is defined by an abstract interface independently of any platform technology. Services are then published in directories where they can be retrieved and used alone or composed with other services. Web services (W3C, 2004) are an important research domain as they
are designed to make the Web more dynamic. Web services extend the browsable Web with computational resources named services. Browsable Web connects people to documents, whereas Web services connect applications to other applications (Mendelsohn, 2002). One goal of Semantic Web services (Fensel et al., 2002; Mc Ilraith et al., 2001) is to make Web services interact in an intelligent manner. Two important issues are Web services discovery and composition, as it is important to find and combine the services in order to do a specific task.

The Semantic Web can play an important role in the efficiency of Web services, especially in order to find the most relevant Web services for a problem or to build ad hoc programs from existing ones.

Web services and the Semantic Web both aim at automating a part of the process of information retrieval by making data usable by computers and not only by human beings. In order to achieve this goal, Web services semantics must be described formally and Semantic Web standards can be very helpful. Semantics are involved in various phases: the description of services, the discovery and the selection of relevant Web services for a specific task, and the composition of several Web services in order to create a complex Web service. The automatic discovery and composition of Web services is addressed in the SATINE European project.
Towards A Meta-Integration Scheme

We have addressed the semantic integration of data in the section 3.2. But as the Semantic Web grows, we now have to deal with the integration of metadata. We have presented ontology merging and ontology mapping techniques in this chapter. In this section, we propose a meta-integration scheme, which we call metaglobal semantic model.

Semantic integration may valuably be examined in terms of interoperability and composability. Interoperability may be defined as the interaction capacity between distinct entities, from which a system emerges. Interoperability in the context of the Semantic Web, will allow, for example, to make several semantic repositories work together to satisfy a user request. On the other side, composability may be defined as the capacity to reuse existing third party components to build any kind of system. Composability will allow building new semantic repositories from existing ones in order to cope with specific groups of users. Automatic adaptation of different components will be necessary and automatic reasoning capabilities are needed for this purpose. This requires a deep understanding of the nature and the structure of semantics repositories. Currently, there is neither a “global” vision nor a formal specification of semantic repositories. The definitions of taxonomies, thesauri, and ontologies, mentioned in the above sections, are still mostly in natural language and, as a paradox, there is not always a computer usable definition of these strategic concepts. This may be the main reason why semantic integration is so difficult to achieve. An effort from the Semantic Web community is needed to provide the Semantic Web community with a meta global semantic model of the data.

Metamodeling for the Semantic Web: A Global Semantic Model

A global metamodel should be provided above data to overcome the semantic repositories’ complexity and to make global semantic emerge. It is important to understand that the goal here is not only to integrate existing semantic objects such as ontologies, thesauri or dictionaries but to create global semantic framework consistency for the Semantic Web. Ontologies, thesauri, or dictionaries must considered as a first level of data semantics; we propose to add a more generic and abstract conceptual level allowing to express data semantics but also to locate these data in the context the Semantic Web.

This global semantic framework is necessary to:
• exhibit global coherence of the data of any kind,
• get insight on the data,
• navigate at a higher level of abstraction,
• provide users with an overview of data space and help them find relevant information rapidly, and
• improve communication and cooperation between different communities and actors.

Requirements for a Metamodel for Semantic Data Integration

A metamodel is a model for models i.e., a domain-specific description for designing any kind of semantic model. A metamodel should specify the components of a semantic repository and the rules for the interactions between these components as well as their environment, for example the others existing or future semantic repositories. This metamodel should encompass the design of any kind of ontology, taxonomy or thesaurus. The design of such a metamodel is driven by the need to understand the functioning of semantic repositories over time in order to take into account their necessary maintenance and their deployments. With this respect, the metamodeling of semantic repository requires to specify the properties of their structure (for example, elementary components i.e. object, class, modeling primitives, relations between components, description logic, etc.). Thanks to these specifications, the use of a metamodel allows the semantic integration of data on the one hand, and the transformation into formal models (mathematical, symbolic, logical, etc.) for interoperability and composability purpose, on the other hand. Integration and transformation of the data is made easier by the use of a modeling language.

Technical Implementation of the Global Semantic Level

The global semantic level could be implemented with a variety of formalisms but the Unified Modeling Language (UML) has already been successfully used in the context of interoperability and composability.

The Unified Modeling Language is an industry standard language with underlying semantics for expressing object models. It has been standardized and developed under the auspices of the Object Management Group (OMG), which is a consortium of more than 1.000 leading companies producing and maintaining computer industry specifications for interoperable applications. The UML formalism provides a syntactic and semantic language to specify models in a rigorous, complete and dynamic manner. The customization of UML (UML
profile) for the Semantic Web may be of value for semantic data specification and integration.

It is worth pointing out that the current problem of semantic data integration is not specific to the Semantic Web. For example, in post genomic biology, semantic integration is also a key issue and solutions based on metamodeling and UML are also under study in the life sciences community.

### Conclusion

In this chapter, we presented a state of the art of techniques which could make the Web more “Semantic”. We described the various types of existing semantic metadata, in particular XML, controlled vocabularies, taxonomies, thesauri, RDF, Topic Maps and ontologies; we presented the strengths and limits of these formalisms.

We showed that ontology was undoubtedly a key concept on Semantic Web pathway. Nevertheless, this concept is still far from being machine-understandable. The future Semantic Web development will depend on the progress of ontologies engineering.

A lot of work is currently in progress within the Semantic Web community to make ontology engineering an operational and efficient concept. The main problems to be solved are ontologies integration and automatic semantics extraction. Ontologies integration is needed because there are already numerous existing ontologies in many domains. Moreover, the use of a common ontology is neither possible nor desirable. As creating an ontology is a very time-consuming task, existing ontologies must be capitalized in the Semantic Web; several integration methods were presented in this chapter. Since an ontology may also be considered as a model for a domain knowledge, the Semantic Web community should consider existing work on metamodeling from the OMG (Object Modelling Group) as a possible way to build a global semantic metamodel to achieve ontology reconciliation.

In the large-scale context of the Semantic Web, automatic semantic integration is necessary to quicken the creation and the updating processes of ontologies. We presented current initiatives aiming at automating the knowledge extraction process. This remains an open research area, in particular the extraction of relationships between concepts. The evaluation of ontology learning is a hard task because of its unsupervised character. In Cimiano et al. (2004) and Maedche & Staab (2002) the hierarchy obtained by applying clustering techniques is evaluated using handcrafted reference ontology. The two ontologies
are compared at a lexical and at a semantic level using lexical overlap/recall measures and taxonomic overlap measure.

The success of the Semantic Web depends on the deployment of ontologies. The goal of ontology learning is to support and to facilitate the ontology construction by integrating different disciplines in particular natural language processing and machine learning techniques. The complete automation of ontology extraction from text is not possible regarding the actual state of research and an interaction with human modeler remains primordial.

We finally presented several research directions which we consider as strategic for the future of the Semantic Web. One goal of the Semantic Web is to provide answers which meet end users’ expectations. The definition of profiles and behaviour models through the combination of Web content and Web usage mining could provide very interesting results.

More and more data-mining techniques involve end users, in order to take advantage of their cognitive abilities; this is the case in visual data mining, in which the knowledge extraction process is (at least partially) achieved through visualizations.

Another interesting application domain for the Semantic Web is the area of Web Services, which have become very popular, especially for mobile devices. The natural evolution of current services is the addition of semantics, in order to benefit from all Semantic Web’s features.

The interest and the need of the Semantic Web have already been proven, the next step is to make the current Web more semantic, with all the techniques we presented here.

References


Section IV

Applications and Policies