Ontology-Based Information Representation

Radostina Georgieva

georgira@in.tum.de

Technische Universität München, Institut für Informatik, Bolzmannstr. 3, D-85748 Garching b. München

Abstract: This paper introduces the concept of ontology and its use for information representation in the context of the Semantic Web project. Ontology is compared to other standards such as taxonomy, thesaurus and topic map highlighting their common and specific features. The existing ontology language OWL is described as part of the Semantic Web project. The framework for Java-based ontologies – Jena – and cluster maps used for visualizing ontologies are introduced. The paper concludes with the role of ontologies in the field of information management.

1 Introduction

The concept of ontology is as old as science itself. Ontology is the science of being. Aristotle believed that the complexity of life could be divided into a natural order based on polar opposites, i.e. dichotomies. For example, Aristotle divided animals into those with blood and those without blood. In the field of computational sciences, ontology is a set of concepts described by terms together with the relationships between them. Relationships carry additional meaning to concepts. They allow viewing only a certain aspect of reality called a domain. Since people view reality differently, aspects of reality are also represented differently. Representations differ in their syntax, semantics and pragmatics [Ca].

Syntax is the set of rules according to which data is combined into other data. For example, one would like to save the sequence of all the web pages visited by a user and be able to check the first one of them. Also, one would like to check whether a user has visited a certain web page at all. The solution to this requirement should be implemented in a language understandable by the browser. As a result, the sequence might happen to be inaccessible from outside the environment it has been created for. The reason is that syntax is devoid of meaning. It describes the structure and the composition of elements and conveys information about their valid order so that they can be decomposed, but does not allow for interpretation. Different standards for data representation use different syntax. Even in the same standard different syntactical representations could be used leading to error-prone or inconsistent.

The meaning of the syntactic elements is provided via semantics. Syntactic elements are like an input to a semantic function that provides some representation of the meaning of the elements as an output. This output might be diverse. Same concept might be named differently, allowing for synonyms and causing redundancy. The semantic function has to give the same output supplied with synonyms. On one hand, semantics quality lowers when important concepts have been neglected, e.g. when there is incomplete data. On the other hand, inconsistent data affects quality as well. Same vocabulary might be used for naming totally different things, known as homonyms, making distinction between them outside a certain context impossible.

Pragmatics is the third area describing concept representation. Different representations of the same concept may be used to achieve various objectives. It is interesting to know in advance which data structure is better suited to solve a certain problem. In order to fulfill the above-mentioned requirement, e.g. to save all the web pages visited by a user, the data structure sequence could be used. It allows a retrieval of its first element as well as a check on whether an instance of a web page is contained in it. However, a stack could do the same just as well as a sequence with the advantage that one could also check what the last page that has been visited. The question of whether this or another data structure is more appropriate is a difficult one. There are many possibilities and certainly not all of them are equally well suited.

Of all the three characteristics semantics is in the center of the current paper. Semantics assigns a meaning to a term and places it in a hierarchy of concepts. Concepts can be specialized or generalized cases of other concepts, can be associated with or be part of other concepts.

Concept modeling is a complex task. It requires good understanding of the meaning of the different concepts. Ontology is a way for successful integration and continuous extension of data models with different syntax, semantics and pragmatics.

2 Semantic Models for Information Representation

Information resources on the web can be viewed as instances of data. Each information resource has a unique address, an owner, a language, a date of last update, a subject. One could view these properties as attributes that allow for an information resource classification. The subject, for example, defines what an information resource is all about. In case its subject is the animal "Jaguar", one could also generalize that it is about a living entity if a hierarchy of living things including animals has been defined in advance. A hierarchy is inherent to biological sciences where organisms are divided into kingdoms, etc.

Taxon	**Term**
Kingdom	Animalia
Phylum (Subphylum)	Chordata (Vertebrate)
Class	Mammalia
Order	Carnivora
Family	Felidae
Genus	Panthera
Species	onca

Fig. 1 Panthera onca [Pa]

Such a classification is possible as soon as the attributes of a set of information resources are expressed by means of terms that are comparable with each other. Another requirement is that the semantics of data is made explicit, e.g. the attributes of information resources are given meaning that is well defined.

An organization of different information resources in the appropriate way allows for the completion of different tasks and is done by using metadata. Metadata is any statement about an information resource, a set of vocabulary about an information resource. It can be used for management of resource's content as well as for discovery of its content – search, navigation, and visualization. The best-known vocabulary for metadata is Dublin Core that defines thirteen properties for describing information resources [TM2]. The property that describes the semantics of an information resource is the subject. It contains a list of keywords. As such, it is insufficient for solving many problems such as, for example, the presence of homonyms, terms that have different semantics but have the same syntax. For example, the term "Jaguar" as an animal and as a car belongs to different domains of knowledge and should therefore also belong to different classes of information resources. The subject, as defined by Dublin Core, is insufficient for describing the semantics of an information resource. Different semantic models for information representation try to solve this problem.

2.1 Taxonomy

Taxonomy is a classification of concepts described by terms according to inheritance (is-a relationship). An example of a taxonomy is the classification of life forms introduced for the first time by the Swedish eighteenth-century naturalist Carl von Linné [TM2]. His taxonomy has been the basis for today's zoological and botanical organism naming system. From this point of view, taxonomy is the science of organizing living things into groups reflecting their natural relationships. Such groups have been called taxa (singular taxon). The names assigned to taxa are collectively referred to as nomenclature.

Inheritance means that a concept can be defined as being broader than or narrower than another concept. For example, living organisms are divided into two disjoint classes based on whether are organism with (eukaryote) or without a nucleus (prokaryote). A mammal is a narrower term than animal but a broader term than, for example, a land mammal.

Also, according to today's classification of organisms, each organism belongs to one of the three domains: bacteria (eubacteria), archaea (archaebacteria) and eukarya (eukaryota). Eukarya includes the kingdom of protista (algae), fungi (saprotroph), animalia (heterotroph) and plantae (autotroph).

A classification could be viewed as the assignment of a concept to a class based on its attributes. An analysis of element used to convey information reveals that they can be displayed as either using graphics or text.

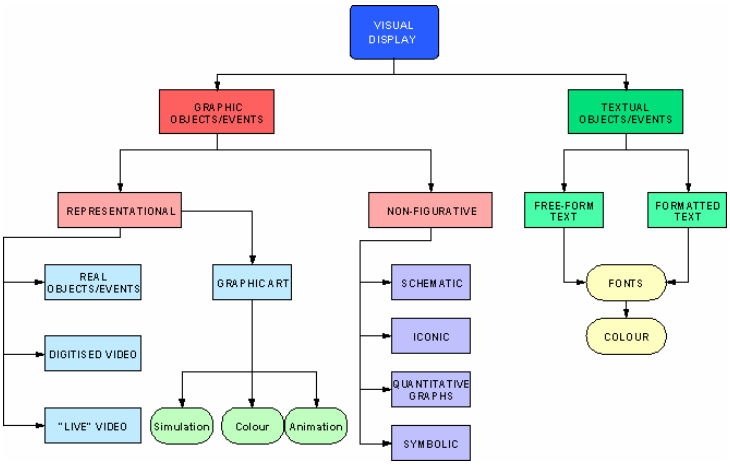

Fig. 2 Taxonomy of visual elements [Hu]

Since there are, in general, many attributes that characterize a concept, taxonomies are too simple to be able to represent the complexity of relationships existing in real life. A disadvantage of theirs is that terms could appear more than once in taxonomy [Ul02]. It happens when a narrower term belongs to more than one broader term.

The following example illustrates inheritance as the basis of a taxonomy:

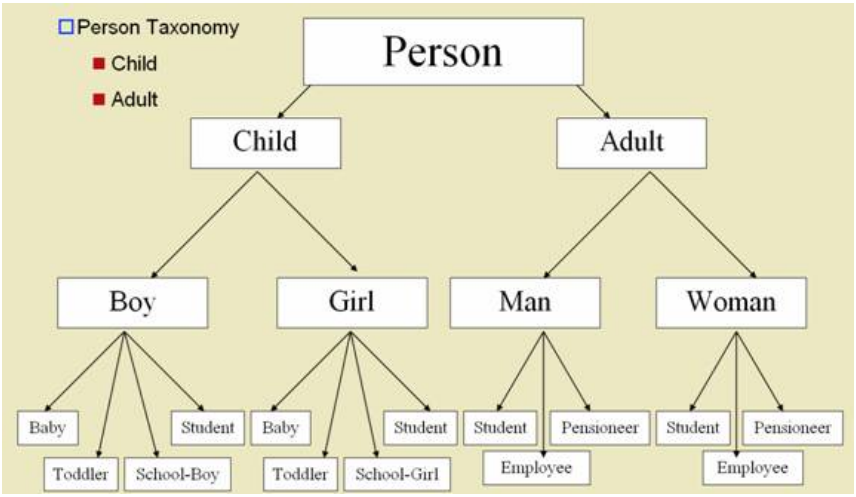

Fig. 3 A taxonomy of "person"

A hierarchy of concepts is represented by terms grouped by the term “person.” According to the rules of this taxonomy, a child is a person and an adult is a person. Also, a boy and a girl are children whereas a woman and a man are adults. However, a baby is either a girl or a boy; a student is either a child or an adult; an employee is either a woman or a man. All of these appear more than once without having a unique identifier. As a result, a taxonomy appears to be a system that lacks formalization. This leads to the introduction of more complex systems for knowledge representation.

2.2 Thesaurus

Thesaurus is defined as a storehouse for knowledge. An encyclopaedia is an example of a thesaurus. It is a classification of concepts defined by vocabulary according not only to inheritance, but also to similarity and synonymy and has been motivated by the librarian sciences and linguistics [TM2].

Guidelines for the establishment and development of monolingual (ISO 2788) and multilingual (ISO 2788) thesauri have been defined by the International Organization for Standardization (ISO) in order to insure consistent practice within a single indexing agency or between different agencies. For example, by means of the thesaurus “Visual Thesaurus” one could look the term “test” up and get related terms, e.g. terms with similar meaning, or synonyms, e.g. terms with the same meaning, building a synonym ring. For example, the terms related to or synonym with “test” are “examination”, “trial”, “try”. Additionally, each term is annotated by a scope note, e.g. a string that explains its meaning. Simple thesauri similar to this one are built in most word processing editors.

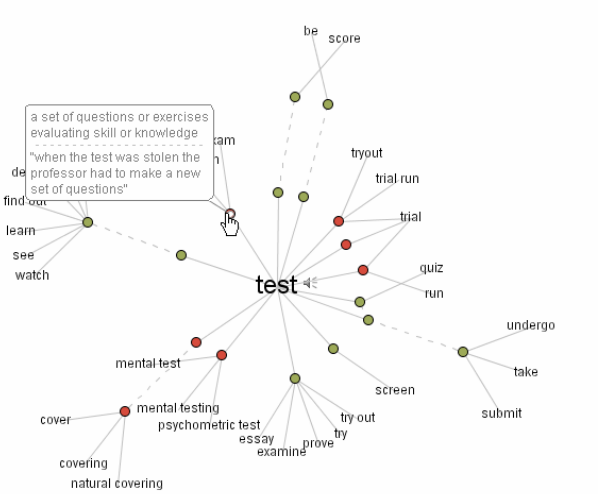

Fig. 3 Thesaurus’ result for “test”. Nouns are marked as green nodes and verbs as red nodes [VT].

A thesaurus would give a hierarchy like this one as a result of a look-up on “child”. For simplicity of representation “child” has been chosen for this example rather than “person”:

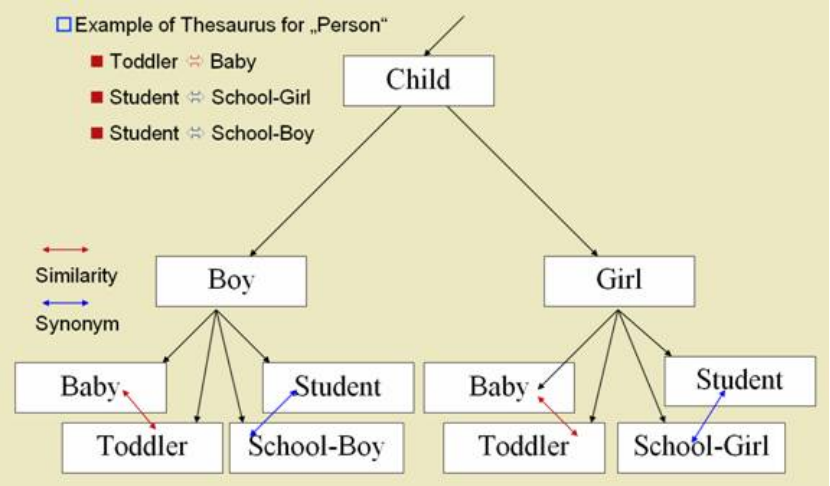

Fig. 4 Thesaurus' result for "child"

In comparison to a taxonomy, a thesaurus extends the hierarchy by adding relationships for similarity and synonymy. However, it does identify terms uniquely and builds on redundancy. Terms could appear more than once in thesaurus when a narrower term belongs to more than one broader term [Ul02].

Also, one term might have more than one meaning. A thesaurus does not make its context explicit. For example, the term "palm" can either mean "any plant of the family Palmae" or "the inner surface of the hand from the wrist to the base of the fingers" or "a unit of measurement of length" or "an award for winning a championship". A thesaurus cannot distinguish between homonyms, e.g. terms with the same syntax and different meanings [Ul02].

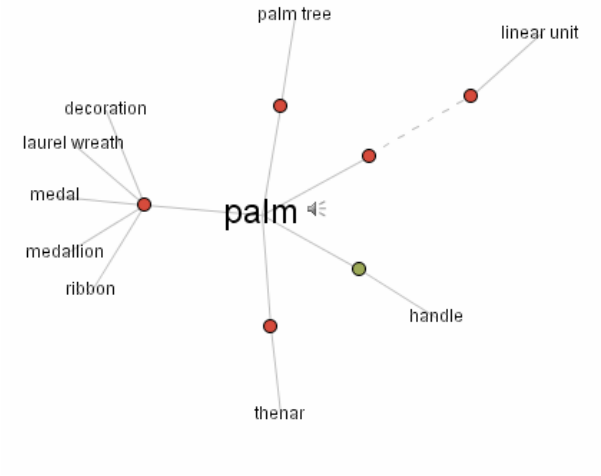

Fig. 5 Thesaurus' result for "palm" [VT]

To overcome these drawbacks, a more complex system for knowledge representation has been introduced.

2.3 Topic Maps

Topic maps are classifications of concepts represented by terms that relate not only based on inheritance, similarity, and synonymy, but also on user-defined relationships. As a model, they have been motivated by the way human's brain works. It creates concepts out of the terms that populate information resources and

uses them for navigation [TM2]. An ISO/IEC 13250 standard based on SGML, a mark-up language that XML is a subset of, describes how a topic map is to be built, without specification of a query language and a unifying data model. Different providers have been implemented topic maps. The most popular among the implementations is the XML Topic Map (XTM), the standard for XML-based topic maps.

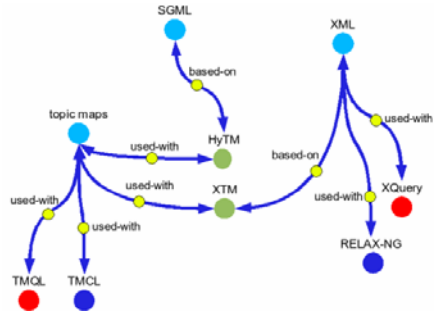

Fig. 6 XML Topic Map [TM2]

Topic maps are based on topics. A topic is an addressable representation of a concept, e.g. a term that is optionally identified by URI. Each topic has one or more names and might have properties that are topics themselves as well. Every topic might be a synonym ring since it might have multiple names. A set of topics builds a topic space. Each topic has a name with a scope and a type, occurrences and associations [TM2].

The name's type defines the range of a topic's name and is a topic as well. Topic maps allow for definition of many different types. The name's scope defines the validity of a topic's name. A topic's name might belong to a scope of an unlimited validity if it is unique. However, the topic "palm" should not be assigned to an empty scope of unlimited validity since there should be some restriction on its use in case it is a tree or a part of the body. On the other hand, an obsolete scope might be assigned to a topic's name that is not recommended for use, for example the topic's name "get off" should be used instead of the archaic or obsolete term "disembark". Additionally, it is possible to define different types of scope [TM2].

Each topic is assigned one or more information resources from outside the topic map that are optionally identified by URIs. Therefore, topic maps are a flat set of terms connecting topics from a topic space and information resources from a resource pool. Resources from the same topic space belong to a resource pool. There relationships between topics and information resources are referred to as occurrences. They might also have scope and type [TM2].

Topics between themselves are connected via associations. Associations are a generalization of the different types of relationships between terms. Associations include inheritance, similarity, synonymy, composition, etc. and needn't be identified by URI [TM2].

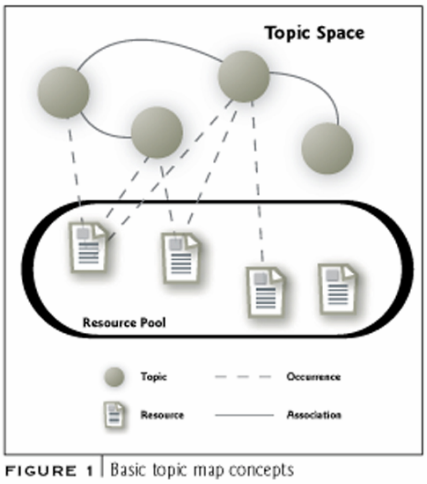

Fig. 7 Basic topic map concepts [TM3]

The following example of a topic map for “person” represents the taxonomy or the thesaurus hierarchy enriched by relationships different from similarity and synonymy:

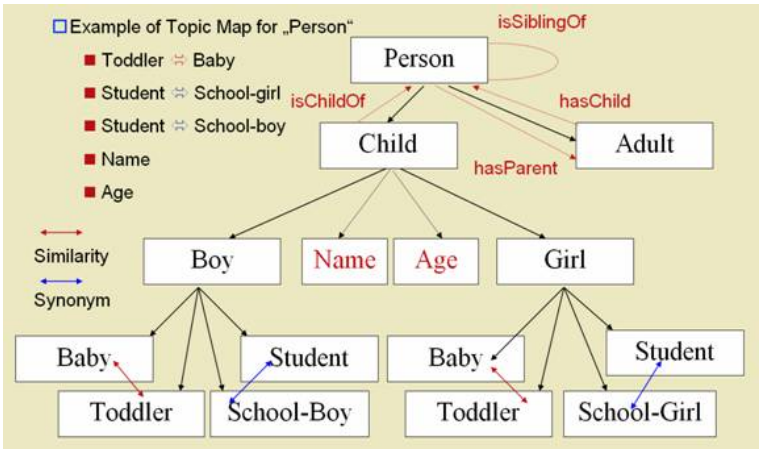

Fig. 8 Topic map for “person”. The topic map for “adult” has been left aside for simplicity.

In comparison with taxonomy and thesaurus, a topic map is much more flexible. It allows for type definition of topics, associations and occurrences. “Palm” is assigned a type “plant”, “body part”, etc. Therefore, the problem of ambiguity is solved by defining different topic spaces. This has an implication for more powerful and precise searches.

However, flexibility has been traded off. In the flat hierarchy inheritance is on the same conceptual level as composition [Ul02]. This is often considered a disadvantage since composition (part-of) relationship is not explicitly defined. Additionally, the redundancy problem remains unsolved.

2.4 Ontology

An ontology might be considered the most complete and powerful model for information representation. It is not only a classification of concepts but also of individuals. In philosophy, ontology studies the basic categories of being trying to find what entities exist and what types of entities exist. In computational sciences ontology is an attempt to define an exhaustive and formal hierarchy within a given domain of knowledge that contains all the relevant terms, the relationships between them and the rules within the

domain. The model consists of a set of types with their properties, with relationships of different types between them, a set of rules and instances [TM2].

Instances belong to broader and to narrower terms. It is possible to distinguish two cases: an exhaustive partition and a disjoint partition. If instances can only be assigned to one subtype, then they belong to a disjoint partition. For example, an individual cannot be both a boy and a girl. If an instance of a term is definitely an instance of one of the partitioned subterms, then the partition is exhaustive. For example, a mammal should definitely belong to an order, a family, a genus and a species. It cannot be just a mammal.

The familiar example for "person" is enriched with semantics leading to the following ontology:

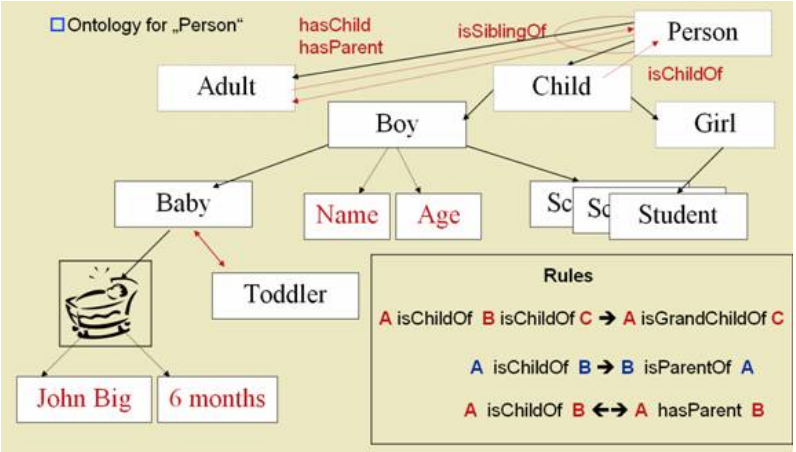

Fig. 9 An ontology for "person"

All of the above mentioned models are appropriate for information management. Taxonomies and thesauri can be used in simple applications whereas topic maps and ontologies are useful for the completion of more demanding tasks. In terms of ontology, a taxonomy is defined as an ontology built on a single type – a term – and a relationship of a single type – inheritance [TM2]. A thesaurus is ontology with a single type but every term has the scope note property and takes part in relationships of altogether three types – inheritance, synonymy and similarity [TM2]. Ontologies are richer since they allow clear differentiation of relationships in terms of their type, e.g. symmetrical, transitive, and inverse. By introducing different types of relationships the elegant and simple tree structure as known in taxonomies becomes complex and significantly more difficult to interpret manually. It is not difficult to understand why. A term might be a part of one term and be a narrower term than another term, e.g. it might have more than one parent. The structure that emerges is known as a directed acyclic graph and it manages redundancy and is easily extensible. To conclude with, ontologies are an investment since they offer extensions in the direction of rule-based systems and integration.

3 The Semantic Web

In order to be useful ontologies need to be expressed in a concrete notation. Ontologies can define rich semantics of complex objects and are therefore well suited for describing heterogeneous information resources. In fact, they are considered to be the key technology to make the Semantic Web become reality.

The Semantic Web has its goal to enrich information resources with semantics. The World Wide Web has defined ways to identify documents uniquely, to access information resources, to classify them. What the Semantic Web has contributed with is that it also allows viewing a document as a machine representation of a certain domain, of a certain aspect of reality. The Semantic Web provides a standard web ontology language and tools and services to help users design and maintain high quality ontologies, store instances, query them and integrate them. Thus, it improves data quality and reduces processing costs and products' time to market.

Specifying an ontology language from scratch, e.g. its syntax, semantics and pragmatics, is a difficult task. Therefore, the Semantic Web has been built on a framework of languages, the Semantic Web Pyramid that has a layered architecture. It defines syntactic conventions and the model for metadata on separate layers of abstraction.

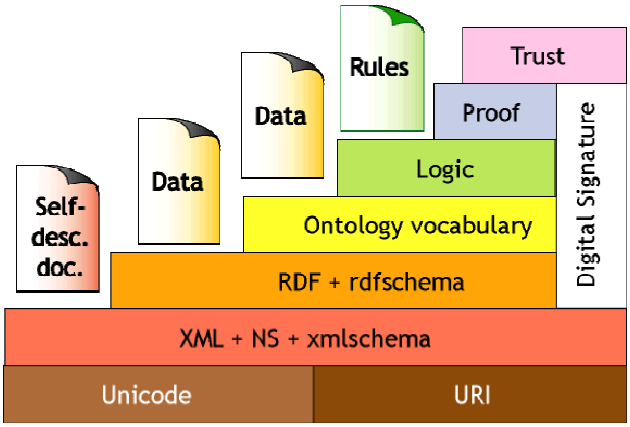

Fig. 10 The Semantic Web pyramid of languages [Sc]

3.1 eXtensible Markup Language (XML) for Data Exchange

One of the core components of the Semantic Web is XML. If the reader is not familiar with it, the online documentation is a good starting point [XML]. In short, it is a markup language that specifies vocabulary for structuring documents by means of data elements and attributes belonging to a certain domain. A well-formed XML document creates a balanced tree of nested open and close tags. The constraints on the order and the combination of the tags are done using document type definition or XML schema definition. XML and DTD or XSD only specify syntactic conventions, e.g. the grammar, and do not define semantics.

XML provides numerous advantages for data exchange. It separates content from form allowing for content to be displayed in many different ways. This is achieved by transformations (rendering) of content using other XML-like languages. Another advantage is its strict syntax and open vocabulary, e.g. users can freeky define datatypes, as well as valid-ability by DTD or XSD. Last but not least is the convention of name spaces – URIs that vocabulary is associated with – allowing for unambiguous interpretation of elements and attributes because inheritance of datatypes, elements and attributes is possible.

3.2 Resource Description Framework (RDF) for Assertions

RDF is an XML-based hierarchy that makes it possible to relate one information resource to one another [Br01]. It is possible to represent relationships using information resources that correspond to real-life entities using assertions. Assertions are the basic building units for encoding information in the RDF model. They consist of an object, an attribute and a value [RDF].

For example, it is known that Jane has a child called John. This relationship can be expressed in A(O,V) format as follows:

isChildOf(http://www.person.bgr/john, http://www.person.bgr/jane)

Additionally, it is also possible to use literals as values. The representation of the assertions “The full name of the person whose information resource has been referred to is John Big” and “John is six months old” in RDF is as follows:

name(http://www.person.bgr/john, “John Big”)

age(http://www.person.bgr/john, “six months”)

Additionally, it is possible to state that an entity is of an element of a set of is of a certain type, e.g. John is a person:

type(http://www.person.bgr/john, http://www.person.bgr/person)

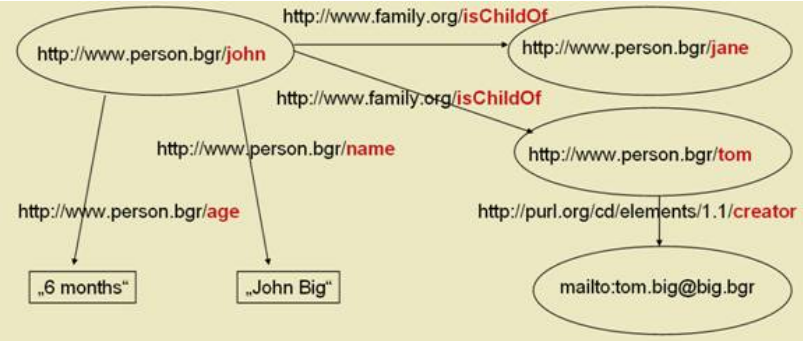

Fig. 11 A graph representation of the example with “John” as an object

```
<Description about=„http://www.big.bgr/john“>
    <person:name resource=„John Big“/>
    <person:age resource =„6 months“/>
    < family:isChildOf resource =„http://www.person.org/jane“/>
    < family:isChildOf resource =„http://www.person.org/tom“/>
</Description>
<Description about=„http://www.big.bgr“ dc:creator=„tom.big@big.bgr“>
</Description>
```

Fig. 12 A RDF syntax of the example with “John” as an object

One of the advantages of RDF is the simplicity of its assertions. As a result, RDF graphs are flexible, easy to handle and chain. One can even express doubt or support of statements expressed by other people. The literature mentions their biggest advantage being the reification mechanism [RDF]. Another advantage of RDF is its ability to use the basic triple model to define meta-data as well as instances of data. RDF is more suitable for exchanging metadata in comparison with XML. In XML, that has not been taken into consideration. However, RDF lacks a mechanism for declaring property names like author or title that are to be used to allow for data exchange between parties (see [RDFS]). Another disadvantage is that the order of elements in the world of metadata is chaotic which creates a discrepancy between the way RDF and XML define information resources (in an orderly manner). This is due to the fact that semantics cannot be completely formalized [Br01]. The possibility to declare properties is defined in the resource schema specification RDFS.

3.3 RDF Schema for Simple Ontologies

RDF Schema provides a basic type system for RDF models [RDFS]. It introduces basic ontological modeling for information resources. RDFS fulfills a different role than XSD. RDFS provides information about the interpretation of an assertion given RDF and does not constraint the syntactical appearance of an RDF description while XSD does.

RDFS defines [RDFS]:

- Classes

- Subclasses that define the hierarchical organization of classes
- Properties
- Subproperties that define the hierarchical organization of properties
- Domain and range restrictions of properties

The type tag is used to specify that an entity is an instance of one or more classes as defined by RDF. The addition of a class tag allows organization within the classes themselves [Ka].

```
<!DOCTYPE rdf:RDF [ <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
        <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'> ]>
            <rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;">
                <rdf:Description rdf:ID="Music">
                    <rdf:type rdf:resource="&rdfs;Class"/>
                </rdf:Description>

                <rdf:Description rdf:ID="Symphony">
                    <rdf:type rdf:resource="&rdfs;Class"/>
                    <rdfs:subClassOf rdf:resource="#Music"/>
                </rdf:Description>
                <rdf:Description rdf:ID="Concerto">
                    <rdf:type rdf:resource="&rdfs;Class"/>
                    <rdfs:subClassOf rdf:resource="#Music"/>
                </rdf:Description>
            </rdf:RDF>
```

Fig. 13 An RDF definition of “Music”

One of the advantages of RDFS is its rich vocabulary with respect to relationships between terms [RDFS]. However, the expressiveness of RDFS is rather limited. Range or domain constraints of a property applicable to two different classes cannot be defined. Cardinality and existence constraints are a difficult issue. Additionally, the available relationship types are insufficient, e.g. there are no definitions for transitive, inverse or symmetrical properties [OWL]. In order to add more semantics to terms, a more complex language has been defined.

3.4 OWL

Representing clear, explicit, machine-understandable semantics is not a trivial task. Different users have different needs. Also, different semantics need different expressive power. Therefore, it is important to have a simple core that suits simple taxonomies and relationships as well as additional layers that allow higher levels of expressivity. This allows also simpler development of inference tools, definition of simple as well as complex ontologies, distribution of the scalability and maintenance burden to the different layers of the language [OWL].

The web ontology language OWL has been designed with this idea in mind. On one hand, it is based on RDF(S) adding expressive power to it. On the other hand, it has three layers of its own.

- OWL Lite is tailored for producing simple taxonomies and ontologies that are easily defined by axioms.
- OWL DL is closest to the predecessor of OWL, DAML+OIL. DL stand for description logics that is a subset of first order predicate logic.
- OWL Full is the richest among the three. It allows representation of predicates of higher order.

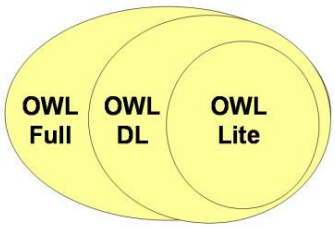

Fig. 13 OWL layered architecture [Sc]

An example of a class definition that RDFS is not mature enough to express is the following definition of "man":

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

Fig. 14 A man is defined as being a person and an adult and being disjoint with a woman

Another example illustrates the definition of how a property relates to other properties:

```
<owl:ObjectProperty rdf:ID=„isChildOf">
        <owl:inverseOf rdf:resource="#isParentOf"/>
</owl:ObjectProperty>
```

Fig. 15 The property "isChildOf" is the inverse of the property "isParentOf"

The biggest advantage of OWL is its adequate expressive power. For example, OWL allows stating that girl and boy are disjoint classes, e.g. no individual can be both a boy and a girl [OWL]. Also, OWL allows stating that the relationship "isChildOf" is the inverse of the relationsjip "isParentOf." Its reasoning capabilities provide consistency checking and check for implied relations. Its reasoning mechanism is based on open world assumption. Regarding interoperability, OWL allows partial definition of mapping rules, sharing, cross linking the inter-ontology relations. Thus, it allows minimization of intellectual effort involved in developing ontology by reusing it. Additionally, OWL provides built-in versioning functionalities [OWL]. Its layered architecture guarantees scalability. Its main drawback is that it is still under development [OWL].

3.5 The Semantic Web Framework Jena

Jena is a Java framework for manipulating ontologies defined in RDFS and OWL Lite. Ontologies might be made available as Java classes and might be cross-linked, shared, queried, etc. The main building blocks of Jena are its RDF API with RDF/XML parser and persistence subsystem, its ontology subsystem, its reasoning subsystem, and the RDQL query language.

The RDF API provides methods for manipulating the set of RDF triples. It also provides a way for extending the behavior of resources. An important part of RDF API is ARP, the Jena's RDF/XML Parser. Another building block of Jena is its persistence subsystem that provides a way to store models using database. In the current release of Jena, Jena2, a de-normalized schema is used allowing for faster insertion and retrieval despite using more storage. However, configuration options are available. The persistence

subsystem supports also a capability for RDQL queries that dynamically generates SQL queries in order to perform as much of the RDQL query within an SQL database engine as possible.

Jena's ontology subsystem's API allows manipulating ontology data based on RDF, such as OWL, DAML+OIL and RDFS. An important component of this subsystem is the document manager that assists with process of managing imported ontology documents. The ontology API works closely with the reasoning subsystem that includes a generic rule based inference engine together with configured rule sets for RDFS and for OWL Lite. It uses inference to derive additional information from a particular ontology source. Reasoners can be used to construct inference models which show the RDF statements that follow reasoning over data has been done. The OWL configuration is still under development.

RDQL query language is a query language for RDF data. Its implementation is coupled to relational database storage so that optimized query is performed over data held in a Jena relational persistence store.

Jena is a comprehensive RDF/OWL Lite framework. It is a complicated system with several different kinds of models and ways of constructing them, e.g. by programming them in Java or reading them as RDF files.

The following example illustrates how one could program a model in Java. Getting an empty model is possible by means of the singleton class `ModelFactory`:

```
Model model = ModelFactory.createDefaultModel();
```

Resources, properties and the statements need to be created explicitly:

```
Resource john = model.createResource(familyURI+"john");
Resource jane = model.createResource(familyURI+"jane");
Property childOf = model.createProperty(relationshipURI);
Statement statement = model.createStatement(john, childOf, jane);
```

Querying is possible using the created model

```
model.listObjectsOfProperty(childOf);
model.listStatements(john,childOf, null);
```

as well as using objects:

- Addition of properies to subjects
  `john.addProperty(childOf,jane);`
- Querying of properties
  `john.listProperties(siblingOf);`

For more examples on how to create and query ontology models, please refer to the tutorial [Jena].

4 Ontology-based Information Visualization with Cluster Maps

Representation techniques discussed so far have used mostly textual elements. Cluster Maps from Spectacle is a visualization technique used for data analysis within a single domain, comparison of different data sets or querying.

A cluster map is a hierarchy built from clusters [Da03]. The cluster map in the Fig. 9 displays a collection of information resources – job vacancies – retrieved after a query on the job description belonging to IT, management and technology. Each small yellow sphere represents an information resource. The green spheres represent a term. Inheritance is represented by directed edges pointing towards the broader terms. Information resources that are classified by the same set of terms are grouped in clusters. Balloon-shaped clusters belong to one term only. Overlapping balloon-shaped clusters speak of information resources shared among more than one term [Ge03].

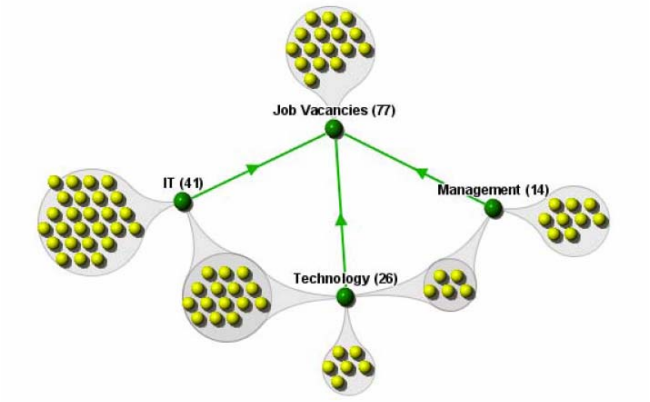

Fig. 16 Job vacancies cluster map [CM]

This example shows the result of data analysis within a single domain. It represents job vacancies distributed according to their job description. The cluster map of 77 job vacancies displays six clusters. Two of them are a result of an overlap. The majority job vacancies of the three are IT-related and are only overlapping with technology-related job vacancies. Additionally, 16 job vacancies have not been assigned to any of the clusters since they belong to an area different from IT, technology and management. This is an example of a partition that is not exhaustive. Cluster maps allow visualization of the overall distribution of information resources. It is visible which objects belong to multiple terms and which terms do not have shared objects. Another aspect of visualization is semantic proximity. Classes that share many instances are semantically close. It is also represented by geographical proximity on the cluster map.

To extract as much information as possible and get a better overview, one could apply a different ontology to the same collection of information resources, e.g. geographical distribution, experience/education, etc [CM].

Cluster maps are also used in data comparison. One and the same ontology is applied to different data sets allowing for their distinction between their characteristics. This can be useful for example for comparison of the services by different institutions [CM].

Another application of cluster maps can be found in querying. Its goal is to find a narrow set of items in a large collection using a conjunction of the terms as a search condition. The result set might be too large or empty. The scope of the search needs to be broadened or narrowed. When there is no search results perfectly matching the criteria it is possible to optimize it. Often there are no exact matches to a user's query and a user gets normally no clear overview of the results and no suggestions for further exploration.
Partial satisfaction of the query is possible by using query relaxation, a technique that allows dropping a criterion and conducting a search with a less restrictive condition. The more relevant the result, the darker is the shade of the color. Using visualization, alternative solutions can be analyzed and presented to the user by neglecting a criterion, for example. Another possibility would be to refine the query by using the ontological nature of the data. A try to approximate one of the subclasses by using its super-class may lead to an interesting query result. On the other hand, if the scope is too broad, a user might choose to replace some classes with more specific subclasses using refinement as the technique that returns a smaller set of options as the scope of the query has been narrowed [Ge03]. In this way, cluster maps provide an overview of the result set even when the conditional clauses are not completely satisfied. For more examples, please refer to [CM], [Da03] and [Ge03].

The advantages of cluster maps over textual data are obvious. The result set conveys information not only in terms of size but also in terms of similarity between the information resources. Analysis, search and navigation are intuitive, insightful, user-friendly and suitable for complicated tasks. Some issues about cluster maps are still open. One issue is making them compatible with RDFS, topic maps, in other words letting them operate on top of Semantic Web repositories.

5 Future of Ontology-Based Information Representation

Ontologies are the most universal way of representing domain descriptions. They are a model for information representation. Information is based on data. However, data could be unstructured, incomplete, error-prone, redundant, distributed, inappropriately scaled, or irrelevant. Therefore, data needs to undergo an analysis and its meaning has to be made explicit in order for information to be retrieved. Information is the set of all the interesting patterns contained in the data. Interesting patterns are those repeating structures in the data that are "generally applicable, not trivial, new, understandable, and useful" [Ru00]. As soon as they are discovered, they can be used in specific contexts. Patterns can be used to represent knowledge specific to a certain domain, to a certain aspect of reality. Because of its specificity, knowledge is often implicit and is therefore difficult to structure, manage and transfer.

However, knowledge needs to be structured, managed and transferred. It is a crucial asset for the survival of an enterprise in today's world where information grows faster than it has ever grown before. An enterprise needs up-to-date, globally distributed, easily accessible information in order to act successfully. The discipline that studies systematic collection, use and storage of information called information management is the basis for the successful completion of many different tasks by users. Surrounded by enormous amounts of data they have to find the exact piece of information they are looking for, to extend a knowledge base with an understanding of its internal structure, to subscribe to the right information source and to get notifications of events they are interested in. The process of information management depends on the available data, the technical resources, the users themselves as well as the organization. In the light of the Semantic Web project and its languages, ontologies suit very well the purpose of building the core of information management of an enterprise.

6 Conclusion

Ontologies have been the main subject of this paper. It has introduced a number of models for information representation starting with the basic ones, the taxonomy and the thesaurus, and concluding with topic maps and the ontologies. These models, in their increasing order of complexity, have been considered for the layered architecture of the Semantic Web. The second part of the paper has presented a short description of each of the Semantic Web Languages, namely XML, RDF(S) and OWL, outlining their main capabilities and strengths. An application of ontologies has been presented in the sections about the Semantic Web Framework Jena and cluster maps. The main goal of the paper has been to present the state of the art of ontology development in connection with the Semantic Web technology. One could conclude that the level of maturity of the languages for ontology specification is high enough, but a lot more effort needs to be undertaken in order to exploit the full potential of ontologies applied in the field of information management. However, the path has already been set by the Semantic Web languages.

References:

[Br01] Bray, T.: What is RDF? http://www.xml.com/pub/a/2001/01/24/rdf.html, 22.02.2005

[Ca] Cameron, R.: Four Concepts in Programming Language Description. http://www.cs.sfu.ca/~cameron/Teaching/383/syn-sem-prag-meta.html. 09.04.2005.

[CM] Fluit, C. et al.: Ontology-based Information Visualization. http://www.cs.vu.nl/~marta/papers/VSW02Book.pdf, 22.02.2005.

[Da03] Davies, J.; Fensel, D.; van Harmelen, F. (Eds): Towards the Semantic Web.John Wiley & Sons, 2003. Chapter 9.

[Ge03] Geroimenko, V., Chen, C. (Eds): Visualizing the Semantic Web, Springer-Verlag London, 2003. Chapters 3 and 4.
[Hu] Hugo, J.: Visual Literacy and Software Design. http://www.chi-sa.org.za/articles/vislit2.htm, 22.02.2005.
[Jena] McCarthy, P.: Introduction to Jena. http://www-106.ibm.com/developerworks/java/library/j-jena/, 22.02.2005.
[Ka] Kanzaki, M.: http://www.kanzaki.com/docs/sw, 22.02.2005.
[OWL] OWL. http://www.w3.org/2004/OWL/, 22.02.2005.
[Pa] Threatened Species. Panthera Onca. http://www.redlist.org/search/details.php?species=15953, 22.02.2005.
[RDF] RDF. http://www.w3.org/RDF, 22.02.2005.
[RDFS] RDFS. http://www.w3.org/TR/rdf-schema/, 22.02.2005.
[Ru00] Runkler, T.: Information Mining. Vieweg Verlagsgesellschaft, 2000.
[Sc] Schreiber, G. The making of a Web Ontology Language a chair's perspective. http://www.cs.vu.nl/~guus/public/2004-webont-zeist/all.htm, 22.02.2005.
[Sw02] Swartz, A. The Semantic Web In Breadth. http://logicerror.com/semanticWeb-long, 22.02.2005
[TM0] XML Topic Maps. http://www.topicmaps.org/xtm/1.0/, 22.02.2005.
[TM2] Garshol, L. Metadata? Thesauri? Taxonomy? Topic Maps! http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html, 22.02.2005.
[TM3] Topic Maps. http://sys-con.com/xml, 22.02.2005.
[Ul02] Ullrich, M., Maier A., Angele J.: Taxonomie, Thesaurus, Topic Map, Ontologie - ein Vergleich. http://www.ullri.ch/ullri_ch/download/Ontologien/ttto13.pdf, 22.02.2005.
[VT] Visual Thesaurus. http://www.visualthesaurus.com/, 22.02.2005.
[XML] XML. http://www.w3.org/XML/, 20.02.2005.