

Master's Thesis

Using XTM for navigating UDDI

Submitted in partial fulfilment for the Degree of a
Master of Science in Information Engineering

at the:

Department of Mathematics and Computer Science,
University of Osnabrück, Germany

and

Department of Computer Science,
University of Twente, The Netherlands

by:

Dana-Roxana Grigoraş
Mecklenburger Str. 4-214
D – 49088 Osnabrück

under the supervision of:

Dr. Paul van der Vet (University of Twente)

and

Diplom-Physiker Thomas Severiens (University of Osnabrück)

Osnabrück, 2005-06-26

for Mom and Dad

Acknowledgments

I would like to express my gratitude for the advice and direction provided by Dr. Paul van der Vet, my first supervisor. The discussions and suggestions from him contributed substantially to this thesis.

My thanks go to my second supervisor Dipl.- Phys. Thomas Severiens, for his help and encouragement to my work. I am also grateful to the leading personalities in the Topic Maps' field: Lars Marius Garshol, Steve Pepper, Jack Park, Bernard Vatant, Kal Ahmed, who helped me with precious hints.

I want to express my appreciation to the complete didactical staff of the *Information Engineering* Master Course, from both universities, Osnabrück and Twente. Special gratitude and consideration I am due to Prof. Dr. Roland Schwänzl, from whom I firstly heard about Topic Maps, who helped me enormously and believed in me.

To Novi, I express my appreciation for his friendship, understanding and continuous support.

Abstract

The discovery mechanisms of Web Services in the UDDI registries are currently quite weak. UDDI provides an interface for keyword and taxonomy based search. The problem is that it has to exist an exact keyword matching. If the consumer searches for the keyword “apartment”, but the provider registered her/his service under “flat”, the result will not be got. Additionally the existing taxonomies might not be appropriate enough, converging to very long result lists that do not really help. The main inhibitor is the lack of semantics. There are some attempts in improving the discovery situation and XML Topic Maps, as a representative of the Semantic Web, could help solving this problem. The present work addresses the stated issue and is, to my knowledge, a new solution. It aims to show how XML Topic Maps can be used as a tool for organizing and representing knowledge out of a specific domain, in our case bioinformatics. This additional semantic layer, expressed in XML Topic Map notation, will bring some benefits to a potential service requestor, who needs to discover the location of an unknown Web Service or resource in general. A prototypical topic map, representing a subdomain of the Gene Ontology, will be developed and with the help of an implementation it will be shown how the knowledge representation can be linked to UDDI. Finally it will be shown how the XTM can be navigated in different ways, in order to point the aim and find a previously unknown resource. XTM will actually be used for building a complete knowledge base; to each term which a potential user may want to navigate through, it will be assigned a number of external information resources. Some aspects have to be considered; the most important – due to the continuous evolution of the Gene Ontology, the XTM document has to be permanently maintained. At the end it was inferred as conceivable referring directly to the location of the WSDL file and doing the job totally without UDDI.

Content

Acknowledgments	3
Abstract	4
Content	5
1 Introduction	8
1.1 Motivation	9
1.2 Problem Definition	9
1.3 Structure of the thesis	10
2 Semantic Web and Web Services Primer	12
2.1 Introduction to the Semantic Web	12
2.2 Introduction to the Web Services technologies	14
2.2.1 Introducing SOAP	17
2.2.2 Introducing WSDL	19
2.2.3 Introducing UDDI	21
2.3 The Semantic Web and Web Services	24
2.4 Summary	26
3 Knowledge Organization	27
3.1 Background	27
3.2 Glossaries	29
3.3 Indexes	29
3.4 Thesauri	30
3.5 Taxonomies	30
3.6 Ontologies	32
3.7 Summary	35
4 XML Topic Maps	37
4.1 History of XML Topic Maps	38
4.2 What are XML Topic Maps	39
4.3 Characteristics	40
4.3.1 TAO of Topic Maps	40
4.3.1.1 Topics	40

4.3.1.2 Associations	42
4.3.1.3 Occurrences.....	43
4.3.2 Name and NameBase	43
4.3.3 Scope and Namespace.....	44
4.3.4 Published Subjects and Published Subject Indicators.....	44
4.4 Merging XML Topic Maps.....	46
4.5 Creating an XML Topic Map.....	46
4.6 Some tools for XML Topic Maps	51
4.7 LTM as an alternative for editing and creating Topic Maps.....	53
4.8 Summary	58
5 Case Study	59
5.1 Introduction to the case study	59
5.2 Biological ontologies	60
5.3 Linking the XML Topic Map to UDDI.....	62
5.3.1 Ontology to UDDI	63
5.3.2 UDDI to Ontology	63
5.3.3 Totally without UDDI.....	64
5.4 Prototypical implementation.....	65
5.4.1 Step 1: Defining the goal.....	66
5.4.1 Step 2: Considering additional prerequisites	67
5.4.1 Step 3: Reusing existing ontologies or defining new ones	67
5.4.1 Step 4: Setting up relevant topics.....	68
5.4.1 Step 5: Assigning information resources	68
5.4.1 Step 6: Setting up associations between the topics.....	69
5.4.1 Step 7: Testing the topic map.....	70
5.5 Navigating the Topic Map	70
5.5.1 Link Navigation	71
5.5.2 Visual Navigation	72
5.5.3 tolog – a topic map query language	74
5.6 XTM in comparison to alternative technologies.....	75
5.7 Summary	76
6 Results	77
References	79

Glossary	87
Appendix A	92
Appendix B	95

1 Introduction

As the biggest library if it is in disorder is not as useful as a small but well-arranged one, so you may accumulate a vast amount of knowledge but it will be of far less value to you than a much smaller amount if you have not thought it over for yourself.

(Arthur Schopenhauer)

Since few decades, the problem of information storage and retrieval has attracted increasing attention. Shortly, the problem can be stated like that: there is a huge amount of information to which accurate and speedy access is becoming more difficult. The relevant information gets ignored, being never sufficiently covered and therefore it leads to a waste of effort and work [Rij96].

Using computers, a fast and intelligent retrieval system is created. But the problem of effective retrieval still remains. The word ‘relevant’ is central in information retrieval context. ‘Reading’ entire documents, either done by humans or computers, presupposes attempting to extract information, syntactically and semantically and deciding whether the document is relevant to a particular request. The problem can be resumed in two directions: *extracting* information and deciding how to use it for being *relevant*.

For humans, it is intellectually possible to ordain the relevance of a document to a query. But for a computer to do this, it is needed to be constructed a model such that the relevance decisions can be quantified. This is the main point with which the information retrieval systems are concerned.

When talking about information retrieval through computers, we touch sooner or later the subject of Semantic Web – this never-ending growing collection of knowledge, built to allow anyone on the Internet to add what they know and find answers to their questions [Haw02]. Its start was a dream: Tim Berners-

Lee, at that time a software engineer at CERN (European Laboratory for High Energy Physics) knew that there had to be a better way to keep track of information about what people were working with. He wrote: *The aim would be to allow a place to be found for any information or reference which one felt was important, and a way of finding it afterwards* [Swa02].

1.1 Motivation

Knowledge is a process of piling up facts; wisdom lies in their simplification.

(Martin H. Fischer)

The need for a new generation of technologies for managing the information flow, data and knowledge is the root of Topic Maps (TMs), a tool which has a heavy word to say in the way of navigating the enormous amount of information provided by the World Wide Web. The Topic Maps concept is the fruit of years of attempts for indexing documents, organizing information and knowledge representation. And because an ontology is a knowledge representation, it plays an important role in Topic Map design [OL02].

It is always a challenge to combine two technologies in order to achieve a better performance. This thesis aims to show how knowledge organisation and knowledge retrieval through XTM (XML Topic Maps) can bring benefits to another exciting area of the Internet, namely Web Services.

1.2 Problem Definition

The strength and flexibility of XTM will be shown in the context of the Web Services discovery with the help of a case study. Web Services are a new and promising topic. Many companies are building their businesses on Web Services technologies. A potential user of these technologies will most probably know the locations of Amazon or Google, but there might be services

of interest, where the location is not known. At this point a potential user would need something like yellow pages in order to find the desired service. In the world of Web Services, UDDI (Universal Description, Discovery and Integration) plays the role of the yellow pages. It is a Web Services registry where service providers register and service consumers discover services. It is important to have such a place, where information about the service and the provider can be found. Now it has to be mentioned that the search mechanism of UDDI is quite limited. However, the discovery process with respect to UDDI could be improved by moving to more semantics. There are many efforts in this area and the aim is to show how to use XML Topic Maps to represent knowledge and benefit in this case from it. The thesis can be viewed as a study that gives an answer to the question *how* this can be realized with XML Topic Maps. Figure 1-1 shows abstractly the goal of the thesis.

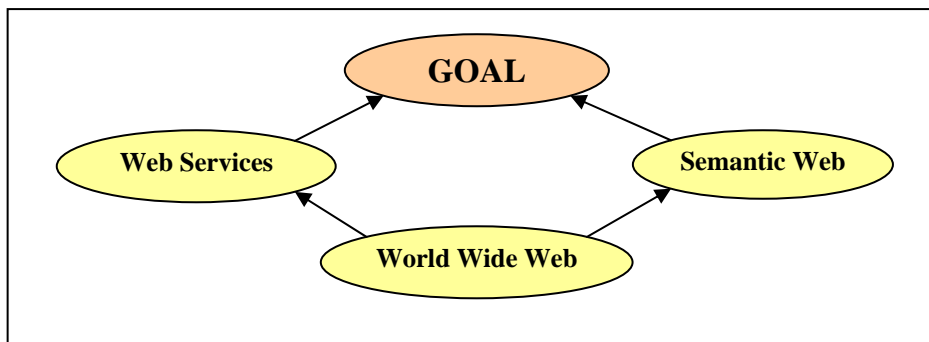


Figure 1-1: The goal of the thesis

1.3 Structure of the thesis

The thesis is focusing primarily on XML Topic Maps and is divided into six chapters. The current chapter acts as an introduction and motivation to the topic. In chapter 2 there will be first introduced the aim of the Semantic Web as well as Web Services and its technologies, because the case study will be concerned in improving the discovery process of services. The problems during the discovery process will be figured out. Chapter 3 deals with the issue of how

to organize knowledge and describes different kinds of expressing and structuring knowledge. In chapter 4 XML Topic Maps will be introduced in detail, beginning with their history and ending with the creation of a sample XML Topic Map. With the theoretical background so far we will move to the case study in chapter 5.

This chapter will deal first with the roots of the Semantic Web and then describe the importance of combining it with Web Services. Finally it will be shown how an XML Topic Map (representing an ontology) can be linked to UDDI with respect to the discovery process. Figure 1-2 gives a graphical overview on the structure of the thesis.

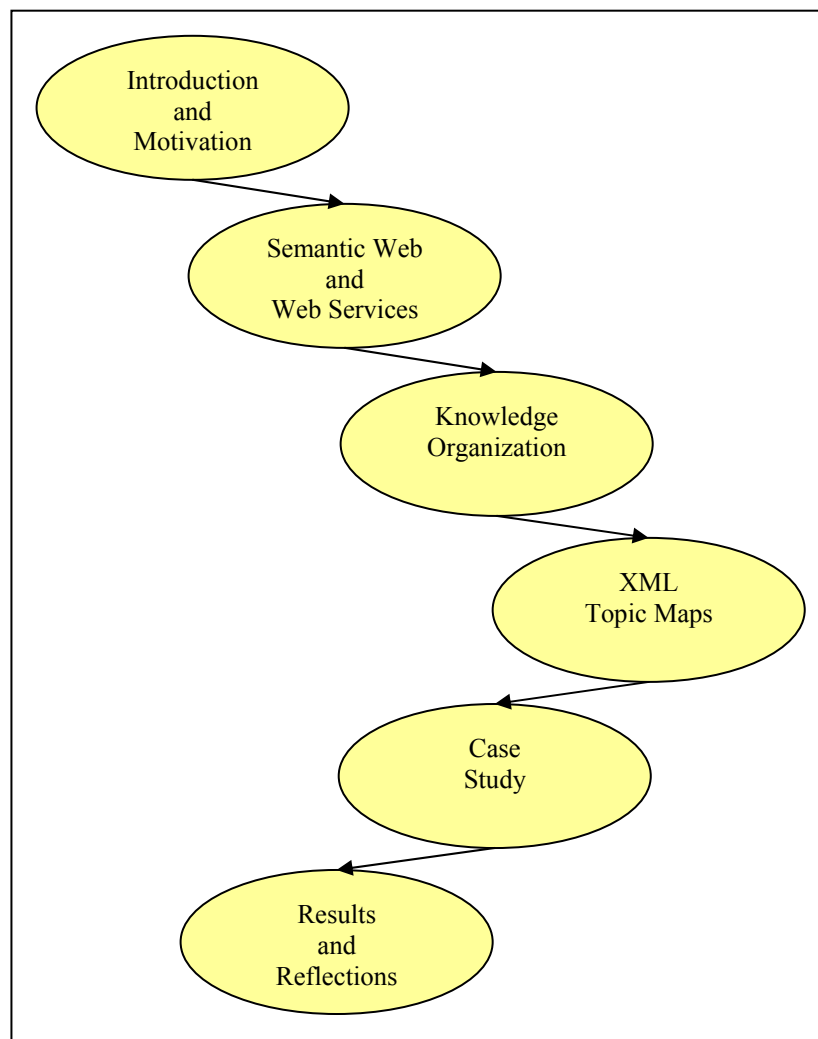


Figure 1-2: The structure of the thesis

2 Semantic Web and Web Services Primer

This chapter provides the basics of two big research areas, namely the Semantic Web and Web Services. The first section gives a brief introduction to the Semantic Web and its roots and technologies. Section 2.2 describes the technologies WSDL, SOAP and UDDI, which built up the concept of Web Services. The following section shows the connection between the Semantic Web and Web Services. The chapter ends with a summary.

2.1 Introduction to the Semantic Web

The World Wide Web has changed the way people communicate with each other, how information is disseminated and retrieved. The term Semantic Web comprises techniques that promise to considerably improve the current web and its use [AH04]. The Semantic Web is rooted by Tim Berners-Lee and is believed to be the next generation of the World Wide Web. Currently the Web is stuffed by meaningless HTML documents so that information can only be retrieved by key-word based searching mechanisms. To make the internet be linked by meaning and semantics rather than hyperlinks, the vision of the Semantic Web community is to develop technologies and mechanisms with increased expressive power and to use them in order to express all resources on the internet. This will lead to a so-called web of meaning.

Ontologies play an important role in the field of the Semantic Web. There exist some description languages that can be used for defining an ontology, like RDF (Resource Description Framework), RDFS (Resource Description Framework Schema) and OWL (Web Ontology Language). Following is a short description of these technologies.

RDF is a language for describing resources and relations between them on the World Wide Web. RDF was designed to provide a common way to describe information such that it can be read and understood by computer applications. RDF descriptions are not designed to be displayed on the web. Information in RDF is structured in triples, which consist of a subject, predicate and object. Each part of an RDF statement can be identified by an URI, so that statements are not only human-readable, but also machine-processable. Although RDF provides a description framework for resources on the internet, it is not capable to define class hierarchy and property hierarchy. It just provides a resource description framework to model the resources without imposing any domain-specific knowledge on it. But this is often needed, as we will see later on in our case study.

To define domain-specific vocabularies, **RDFS** (RDF Schema) was built on top of RDF, and provides a type system for RDF. It provides the facilities needed to describe domain-specific class and hierarchy properties and therefore enable the representation of taxonomies and ontologies.

OWL became a W3C recommendation in February 2004. It is derived from the DAML+OIL Web Ontology Language and is based on XML and RDF. *OWL and RDF is much of the same thing, but OWL is a stronger language with greater machine interpretability than RDF* [OWL05]. In short, OWL is a language for publishing and sharing ontologies and their associated knowledge bases on the web [Liu04].

The main modelling primitives of RDF/RDFS concern the organization of vocabularies in typed hierarchies: subclass and subproperty relationships, domain and range restrictions, and instances of classes. But the expressive power of RDF/RDFS is limited, some features are missing, and there comes OWL and extends RDFS; among them [AH04]:

- Local scope of properties: in RDFS one cannot declare range restrictions that apply to only some classes. For example, one cannot say that elephants eat only plants, while other animals may eat also meat.
- Disjointness of classes: one cannot say in RDFS that *male* and *female* are disjoint, just that they are subclasses of *person*.
- RDFS does not allow combinations of classes using union, intersection or complement; for example to say that *person* is the disjoint union of the classes *male* and *female*.
- Cardinality restrictions: in RDFS one cannot say, for example, that a person has exactly two parents.
- Special characteristics of properties (like *transitivity*: “less than”, *uniqueness*: “mother of”).

In contrast to the mentioned technologies, XML Topic Maps are according to the W3C officially not part of the Semantic Web, but according to the Semantic Web community in general, they are. UDDI is also not a recommended standard by the W3C, but by another organisation OASIS (Organization for the Advancement of Structured Information System). Chapter 4 will show that XTM can be used straightforward for representing domain-specific knowledge. It represents an alternative standard for applying semantic meaning to electronic resources.

2.2 Introduction to the Web Services technologies

Web Services seem to be the next major frontier in computing. The subsections of this section are giving a brief overview on them. Till the advent of Web Services, interoperability and integration (the exchange of data among computer systems) were extremely limited or cumbersome. Prior to Web Services, limited integration took place with numerous technologies, vendors, obstacles, and formats that built a high barrier in sharing of data. But then, Web Service technology came along and changed that [Man03]. Web Services

are programmable and platform-independent; the principle of client server is even older. The novelty consists in the usage of XML. There are many different ways of defining Web Services and two of them will be listed below:

“A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed with HTTP using an XML serialization in conjunction with other XML-related standards [WS04].”

“The XML based Web Services are emerging as tools for creating next generation distributed systems that are expected to facilitate program-to-program interaction without the user-to-program interaction. Besides recognizing the heterogeneity as a fundamental ingredient, these web services, independent of platform and development environment can be packaged and published on the Internet as they can communicate with other systems using the common protocols like HTTP and SOAP [WS02].”

But what exactly are Web Services about? This question can be answered with the help of the Web Services architecture, where three roles can be distinguished, namely the service provider, the service requestor and the service registry. This is illustrated in Figure 2-1.

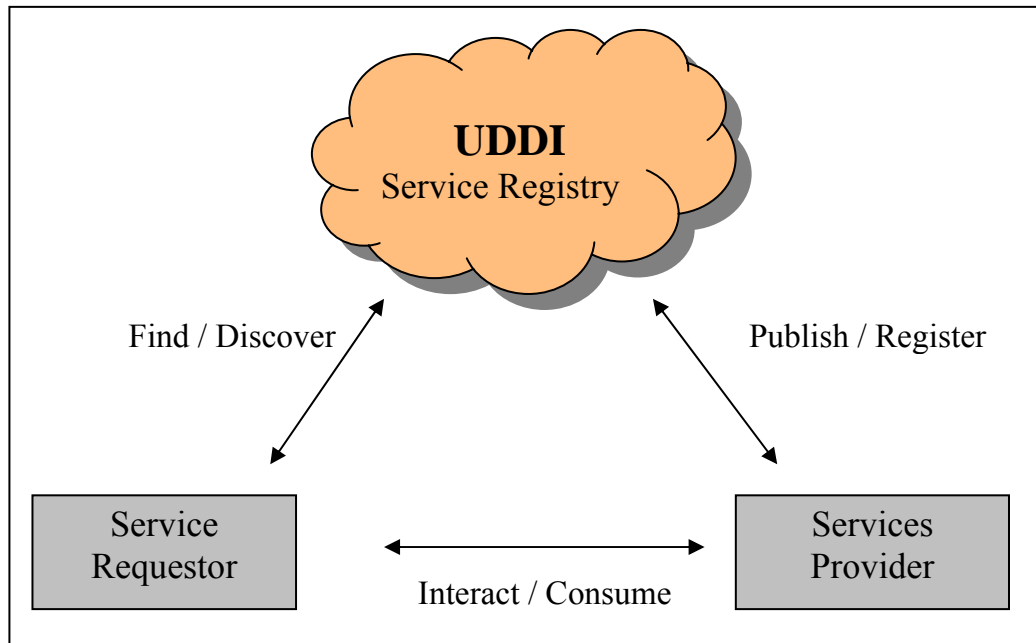


Figure 2-1: The Web Services architecture

The service provider offers a service usually on the internet, but it is also conceivable that a service is offered on the intranet or extranet (the intranet is a private network, contained within an enterprise; the extranet is a private portion of the Internet). The service requestor is looking for such a service and can find information on how to bind to the service and invoke the desired service by addressing the last role in this architecture, namely the service registry (sometimes called discovery agency).

Figure 2-1 displays also the interactions that happen in a Web Service scenario. These operations take place between the mentioned roles and are described below:

- **Publish/Register:** The services provider publishes information about services to the discovery agency, after a description was created.
- **Find/Discover:** The service requestor searches and hopefully finds services for her/his individual needs in the service registry.
- **Interact/Consume:** After the service requestor found the service in the

registry, she/he has the information and ability for interacting (sometimes also: contacting, binding, consuming, etc.) with the service [ZTP03].

Of course, for realizing above operations some technologies have to be applied. When talking about Web Services always three key-terms are mentioned, namely SOAP, WSDL and UDDI. These are XML-based technologies and represent the three core technologies of Web Services. The following subsections describe them briefly.

2.2.1 Introducing SOAP

SOAP is an XML-based protocol for the communication in a Web Services environment. SOAP messages enable the communication between the previous mentioned roles. From the requestor's point of view, SOAP is needed for the discovery of a service in a registry as well as for invoking the service afterwards. HTTP and XML solve the problem of interacting the interfaces between the various platforms in a network, because they are both platform-independent. SOAP explains what data should be in the http-header, as well as what data should be in an XML body of the soap environment of a http message, so that an application in one computer can call an application in another computer and transfer information (data, etc.) between the two. All information needed by a Web server that works with the SOAP protocol is contained in a SOAP message, which consists of a SOAP Envelope, a SOAP Header and a SOAP Body. Figure 2-2 shows the overall structure of a SOAP message.

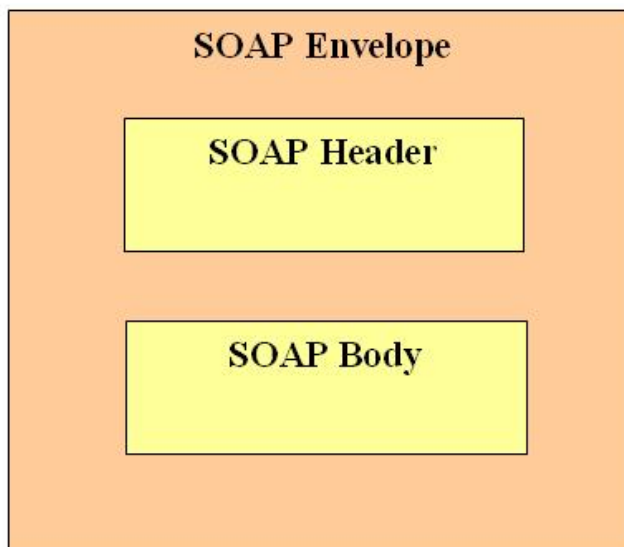


Figure 2-2: SOAP message structure

The envelope represents the entire package of data that is being used to explain how information should be transferred. As part of this envelope, the optional header can contain information about the routing and the delivery options for the SOAP message. The obligatory body, also as a part of the envelope, contains the actual data. Usually it is the method or operation one is invoking [Man03].

The following example is showing the basic XML structure of a SOAP request message. The message starts with the required envelope tag. The namespace <http://schemas.xmlsoap.org/soap/envelope/> is identifying the SOAP version used in the message. Suppose the requestor is interested to get the monthly price for a specific flat, which is identified through an ID.

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/" >

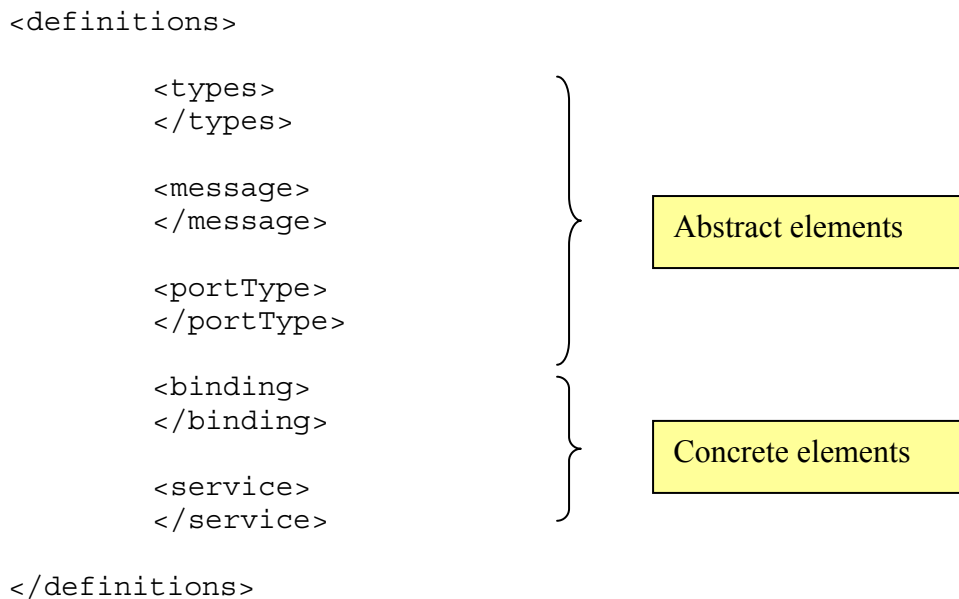
  <SOAP-ENV:Body>
    <getPrice xmlns="http://prices.org">
      <FlatID>10000234</FlatID>
    </getPrice>
  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Further information about SOAP can be found at <http://www.w3.org/TR/2003/RECsoap12-part0-20030624/>.

2.2.2 Introducing WSDL

WSDL stands for Web Services Description Language. It is an attempt to describe Web Services and became the *de facto* standard for this purpose. This is also reflected by the fact that a standardization organization like the W3C is concerned with WSDL, but also with the previously mentioned SOAP. Usually the service provider creates a description about the offered service during the development of a service. For a service requestor it is important to know where a service is located. Further he needs to know what kind of messages the services will understand and what kind of operations are offered, as well as how the messages are encoded and which protocol is used for exchanging the messages. This information can be provided to the requestor by the WSDL file of a service. The following is showing the basic XML structure of a WSDL document:



The `<definitions>` tag acts as a container for the other elements. WSDL separates a service description into **abstract elements** and **concrete elements**. The elements `<types>`, `<message>` and `<portType>` are describing the interface of a service in an abstract way. The `<types>` element provides a container for data type definitions. The `<message>` element is an abstract description of the data that is send. It can be thought of it as outlining the structure of a SOAP body. Remember the service from section 2.2.1, where a requestor was interested to get the monthly price for a specific flat. The WSDL file must contain in this case two different `<message>` tags, one for the request and one for the response. The request and response messages make up the so called request-response operation. This operation will be called *getPrice*, and will be contained within the `<portType>` tag. This shows the location where the consumer can find the method *getPrice* from the previous example and this could look like following:

```
<portType name="PricePortType" >
    <operation name="getPrice" >
        <input message="tns:getPriceRequest"
            name="getPriceRequest" />
        <output message="tns:getPriceResponse"
            name="getPriceResponse" />
    </operation>
</portType>
```

The elements `<binding>` and `<service>` are describing the implementation of the service and are called concrete elements. The `<binding>` element maps a `<portType>` to an existing service implementation. It provides information about the protocol and the concrete data formats expected. The `<service>` element of a WSDL document may be the starting point for a consumer for exploring a service description, because it contains a `<port>` element that describes the service network address, i.e. the location of a Web Service [ZTP03].

Note that it is the duty of the service provider to create a “good” description of the offered service and to publish information out of it and its location to the service registry. Detailed information about WSDL and its elements can be found at <http://www.w3.org/TR/2001/NOTE-wsdl-20010315> .

2.2.3 Introducing UDDI

The UDDI initiative was brought to life by IBM, Microsoft and Ariba. In 2002 UDDI was moved to OASIS, an official organization for standards. It actually initiated from the e-business community.

As it was mentioned earlier, it is a challenge for a requestor to discover and find a service of potential use. The UDDI registry is, let us say, a centralized Web Services search engine helping the service consumer to find adequate service offerings. In a UDDI registry consumers may find:

- Information about businesses and organizations offering Web Services
- Descriptions of the Web Service(s) that these organizations provide
- Information about technical interfaces to these Web Services [ZTP03].

UDDI represents the functionality of a given business entity in terms of a set of data structures. The structures are represented in XML when messages are exchanged. The UDDI specification defines five such structures or elements. Note that each UDDI element can uniquely be identified through the value of a key attribute, namely UUID (Universal Unique Identifier). The UUID is a 128-bit value that is either guaranteed to be unique or extremely likely to be unique until 3400 A.D. [AM03]. It is a hexadecimal string, which is automatically generated by the UDDI server. The UUID string is a dash-separated character sequence of format “8-4-4-4-12”, where the numbers

represent the occurrence of hexadecimal numbers [ZTP03]. The relationship between these structures can be seen in Figure 2 - 3.

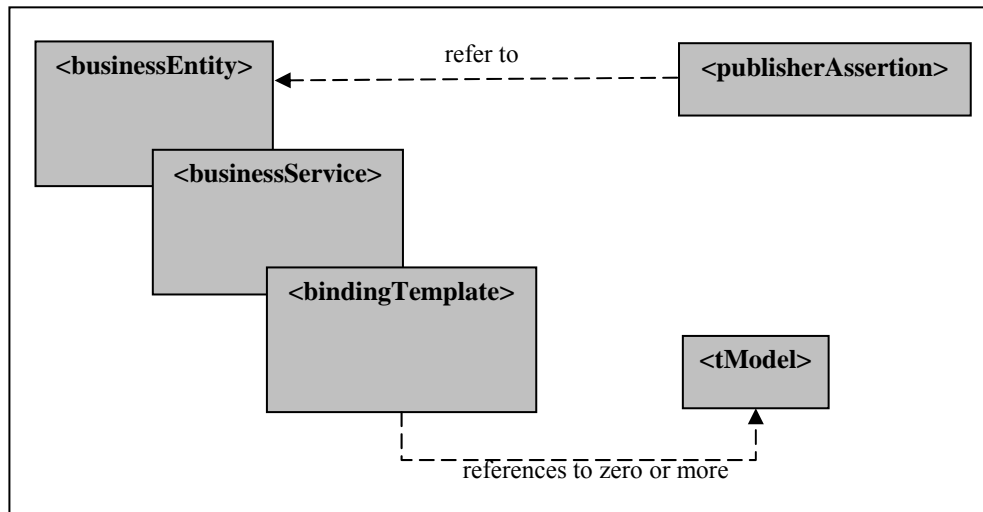


Figure 2 - 3: The relationships between the UDDI data structures

- The **<businessEntity>** structure is the encapsulating document for a UDDI document that describes a business. Within it are informational elements such as name and description, as well as nested instances of **<businessService>**.
- The **<businessService>** element describes a single Web Service provided by the **<businessEntity>** containing it. Within it are again informational elements as name and description, as well as nested instances of **<bindingTemplate>**.
- The **<bindingTemplate>** structure is where the technical descriptions of services begin to take form. Within this structure will be contained references to **<tModel>** structures, as well as the first element directly recognizable as being related to a Web Service, namely the **<accessPoint>** element, which contains attributes defining the means of access to the service itself (URL, email, etc.)
- The **<tModel>** element is present by reference and not contained in the

document itself, like shown in Figure 5.2. The <tModel> structure describes and references abstract data, ranging from business tax codes to categorical identifiers, for instance a geographical classification. On the other hand, a <tModel> can be used for referencing to the WSDL document of a service.

- The <**publisherAssertion**> asserts a relationship between two business entities. The reasoning behind this is to allow large companies to publish UDDI for their subentities [RK02].

Web Services consumers must be able to identify information entities, but they also must be able to select entities according to distinct categories. In particular, when looking for a previously unknown service, a search may be based on distinct categories, like the region where the provider operates. A category bag may complement the elements <businessEntity>, <businessService> or <tModel> and add a categorization to one of the mentioned entities [ZTP03].

For a service provider UDDI offers a Publish API that allows populating the above described elements with information and for a service consumer UDDI offers an Inquiry API for finding that information.

Figure 2-4 shows the Search Web Interface to the UDDI registry, provided by the operator Microsoft. All operators offer UDDI version 3 as a beta version. Therefore the thesis will deal only with UDDI version 2. More information about UDDI can be found at <http://www.uddi.org>.

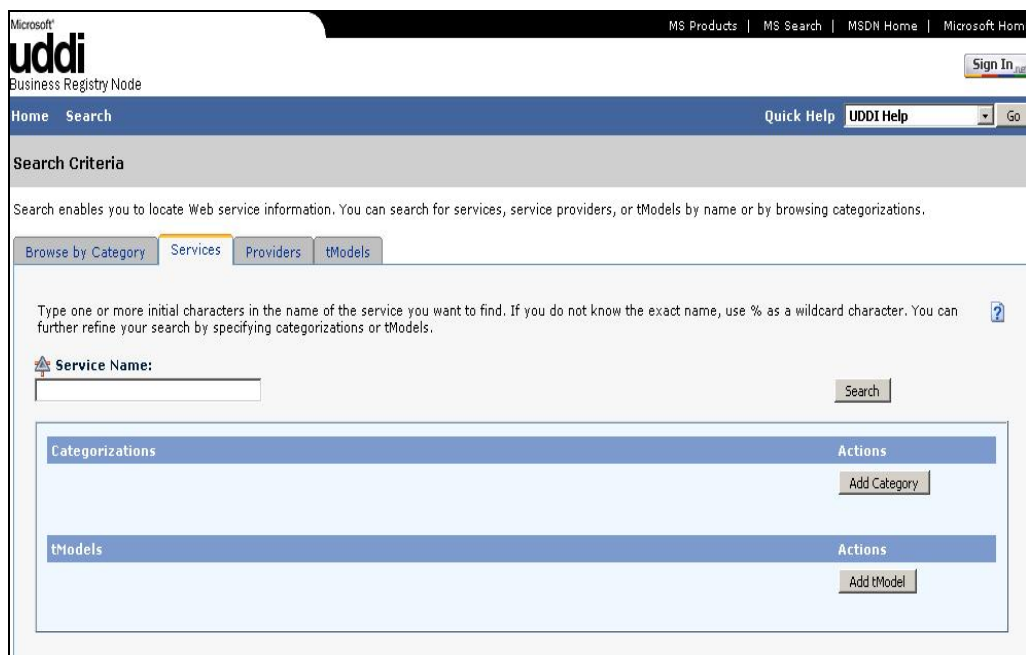


Figure 2-4: Search page of the Microsoft UDDI registry

The UDDI search mechanism has some “lacks”. [SVSM03] describe that the current discovery mechanism supported by UDDI is not powerful enough for automated discovery. The main inhibitor is the lack of semantics/language structure. This makes UDDI less effective, even though it provides an interface for keyword and taxonomy based searching. The problem is that it has to be an exact keyword match. If the consumer searches for the keyword “apartment”, but the provider registered her/his service under “flat”, he will not get a result. On the other hand, if the consumer uses taxonomy for his search, for instance geographical information about the service, she/he will most likely get a result list which does not really help, being usually too long. The case study in chapter 5 will deal with this problem and some possible solutions, with the aim to improve the discovery process, will be presented.

2.3 The Semantic Web and Web Services

The main reason why the technologies Semantic Web and Web Services moved together in the past is to bring the Web to its “full potential” [Fen04]. In

the previous section the discovery problem with respect to UDDI was figured out. It was also mentioned that ontologies have been identified as the solution for this problem. As a result we can say that searching for Web Services could be based on Semantic Web technologies, like XML Topic Maps. There exist some efforts in this direction that use some of the technologies described in the earlier section.

Ivan Herman [Her04] from the World Wide Web Consortium says that the Semantic Web is in the URI-s and that each resource on the web can be identified through an URI. This could be e.g.:

- URL-s
 - <http://www.utwente.nl>
 - <ftp://mathematik.uos.de>
- URN-s
 - <urn:nbn:se:uu-diva-3475>
 - urn:lsid:ensembl.org:homo_sapiens_gene:ensg00000002016.

The semantic problem with respect to UDDI was identified also by the UDDI community. UDDI's features and taxonomies do not always meet customer requirements. [Vos05] says in a discussion paper of January 2005 that each UDDI entity could also be seen as a resource with metadata on the internet. This is visualized in Figure 2-5.

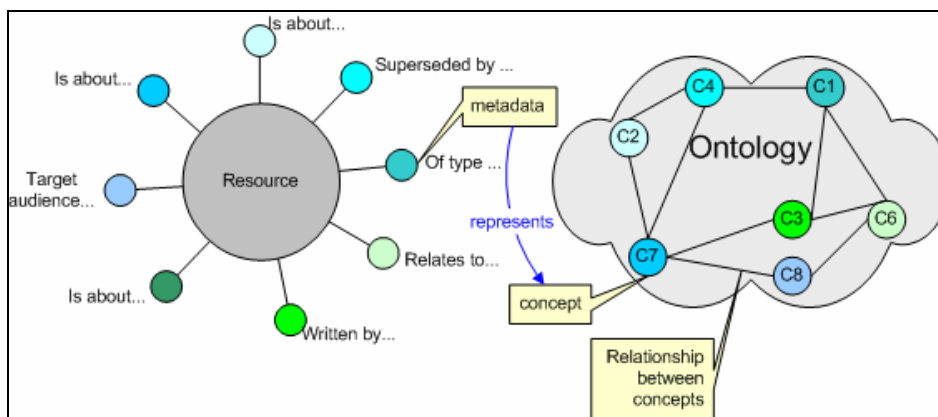


Figure 2-5: UDDI entity as a resource with metadata [Vos05]

The previous figure shows that some metadata of an UDDI entity can represent a concept of an external knowledge representation. There are a couple of ways conceivable how to link a UDDI entity to such an ontology. Some possible options will be discussed in chapter 5.

2.4 Summary

Chapter 2 gave an introduction to the areas of interest. Web Services technologies have been described and brought in connection to the Semantic Web.

Ontologies have been identified by the Semantic Web community as the basis for semantic annotation that can be used for discovery. Therefore the next chapter is devoted to the different principles of knowledge organization, among others, ontologies.

3 Knowledge Organization

When talking about information, one talks implicitly about Knowledge Organization. The main preoccupation in Knowledge Organization is to optimize the organization of the knowledge repositories in order to facilitate the creation, retrieval and share of knowledge.

The field of Knowledge Organization deals with the principles of classification, thesauri and ontologies, as it will be seen in the following sections. The same principles are also the focus of Topic Maps. Knowledge Organization is however, a broad and abstract context for XML Topic Maps.

3.1 Background

What actually is knowledge? The terms *data*, *information* and *knowledge* are usually used in ill-defined ways, which is acceptable in colloquial conversations where only rough, intuitive notion of their distinctive meanings are needed [OL02]. But there is a difference between data, information and knowledge, as it can be seen in Figure 3-1.

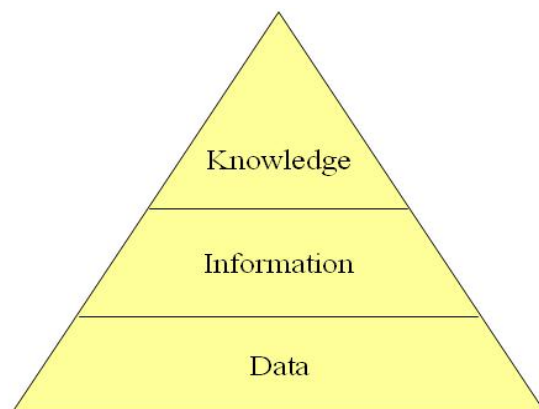


Figure 3-1: From data to knowledge

The concepts of data, information and knowledge are closely related. Data is the raw material of information. It may be facts, events or statements. A single piece of data has no meaning unless the context is understood. Information consists of any tagged data, which is treated as an attribute. Knowledge is the relationship between the attributes [Moo].

Knowledge Organization is about organizing objects of thought (and associated carriers of information) so that humans can work with them more easily. It is the interdisciplinary field that theoretically reflects the practical activity of organizing knowledge for specific purposes and discourse communities. The central aim is improved access, more sophisticated finding aids, and a clearer overview. Knowledge Organization handles with structured metadata (like indexing), expresses and orders statements about subject matter, which are comprised of concepts and relations [Sig02].

There was always a need for systematic representation of knowledge. The field of Knowledge Organization is quiet old. The root can be found in the uncountable years of experience from the area of information science and library. Anderson [And97] defines Knowledge Organization as if it were centered only on documents, instead of knowledge structures. According to him, Knowledge Organization is:

the description of documents, their content, features and purposes, and the organization of these descriptions so as to make these documents and their parts accessible to persons seeking them or the messages that they contain. Knowledge organization encompasses every type and method of indexing, abstracting, cataloguing, classification, records management, bibliography and the creation of textual or bibliographic databases for information retrieval.

3.2 Glossaries

A glossary is a list of specialized words with their definitions, placed usually at the end of the book. It does not point to the occurrence of the topic, as the index does, but it just gives the definition. Sometimes glossaries contain additional information, like *see* and *see also* references or give guidance about language use or pronunciation [CN02].

3.3 Indexes

Index comes from the Latin *index*, which means informer or sign. In a usual dictionary, it denotes an alphabetical list of subjects, names, with references at the end of the book.

We anticipate the next chapter and note that in the Topic Maps context, an index can be seen as a list of topics. It is always a map of the knowledge of the information resource, normally presented in alphabetical order (names of topics) and with reference to those 'topics' (occurrences). In a more complex index, typographical conventions can be used to distinguish between different types of topic (e.g. bold) or even different types of occurrences (e.g. italic) [Oli00].

As additional features of indexes: multiple indexes distinguish topic types, as typographical conventions do, while using explanatory labels following the names makes the distinction between topic types (for instance Paris - city and Paris – the famous Greek).

3.4 Thesauri

A thesaurus is a network of interrelated terms in a specific domain. Given a certain term, the thesaurus indicates which terms denote one related to it, as well as which are related in another way. According to the ISO 2788,

a thesaurus is the vocabulary of a controlled indexing language formally organized so that the a priori relationships between concepts – (for example as ‘broader’ and ‘narrower’) are made explicit.

As dictionaries give to the user information about unfamiliar concepts, thesauri provide the help to find a particular word for which the concept already exists in mind.

3.5 Taxonomies

Taxonomy derives from the Greek *taxis* meaning arrangement or division and *nomos* meaning law and it is the science of classifications according to a predetermined system, with the resulting catalog used to provide a conceptual framework for discussion, analysis, or information retrieval [Tax03].

The creation of comprehensive taxonomies or classifications dates back to ancient times. Aristotle (384-322 b.C.) emphasized strongly classification and categorization schemes. His *Organon*, a collection of works on logic assembled by his students after his death, included a treatise called *Categories* in which he attempted to construct what we would now call an upper ontology. He also introduced the notions of *genus* and *species* for lower-level classification, although not with their modern, specifically biological meaning [RN03].

Our present system of biological classification, including the use of “binomial nomenclature” (classification via genus and species in the technical sense), was invented by the Swedish biologist Carolus Linnaeus, or Carl von Linné (18th century). The problems associated with natural kinds and inexact category boundaries have been addressed by Wittgenstein, Quine, Lakoff and Schwartz (20th century) [RN03].

In a taxonomy, each term is in parent-child relationship to other terms. It may be that they are different types of such relationships, like type – instance, whole – part or genus – species. It is desired that the relations to a single parents to be of the same type. Some taxonomies allow heterarchy, that is, a term has multiple parents. So, if a term appears in taxonomy in more than one place, then it is the same term.

Note that in some situations, one can treat an ontology as taxonomy, and this is actually sometimes done. Both are essentially classifications. The concepts which compose a domain are divided in different classes. Looking at the relationships between concepts, we can derive the meaning of the individual concepts. If one wants to make a rigorous ontology, one makes first a taxonomy.

Taxonomies have been used in technical fields. For example, systematic biology aims to provide a taxonomy of all living and extinct species; library science has developed a taxonomy of all fields of knowledge, encoded as the Dewey Decimal system; and tax authorities and other government departments have developed extensive taxonomies of occupations and commercial products. Taxonomies are also an important aspect of general common-sense knowledge [RN03].

3.6 Ontologies

In this section, ontologies will be discussed first briefly and then some aspects of this formal structure will be adopted, in the context of Semantic Web, in order to provide a better understanding for the following approach of the later chapters.

There are many ways of defining ontologies, differing according with the domain in which the term is wanted to be analyzed. Philosophy, Information Technology, Artificial Intelligence, Topic Maps are only some of the possible fields which present ontologies in a relatively different manner. We nominate in the following few of them, culminating with what we want to emphasize and point on.

Ontology [CN02] is a discipline of Philosophy that deals with what kinds of things exist - which entities there are in the universe. It is a branch of metaphysics, the study of first principles or the essence of things. It derives from the Greek *onto* (being) and *logia* (written or spoken discourse).

Leo Obrst and Howard Liu [OL02] describe ontology as including:

- *Entities* (things)
- The *relationships* between those entities
- The *properties* (and property values) of those entities
- The *functions* and *processes* involving those entities
- *Constraints* on and rules about those entities.

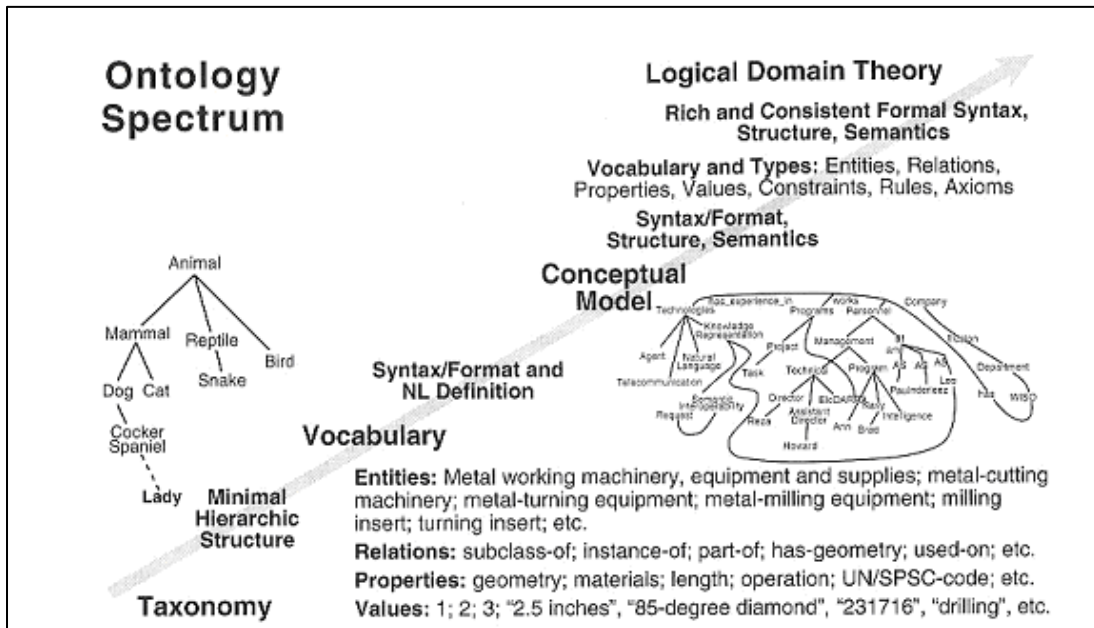


Figure 3-2: Ontology spectrum by Leo Obrst and Howard Liu [OL02]

As shown in Figure 3-2, the meaning of an ontology can range from the notion of a taxonomy (knowledge with minimal hierarchical structure) to a vocabulary (machine-readable knowledge as standardized terminology with natural-language definitions) and up to a conceptual model (with more complex knowledge representation), concluding finally in the notion of an ontology as a logical domain theory (with very consistent and meaningful knowledge). So, an ontology acts as a semantic conceptual model representing common knowledge in a well-defined, extensible and modular fashion [OL02].

Recently, ontologies have moved from a topic in philosophy to a topic in applied artificial intelligence that is at the centre of modern computer science. Tim Berners - Lee, Director of the World Wide Web Consortium, referred to the future of the current WWW as the *Semantic Web* – an extended Web of machine-readable information and automated services that extend far beyond current capabilities. *Ontologies* are the backbone technology for the Semantic Web [Fen04].

Ontologies provide a shared understanding of certain domains that can be communicated between people and application systems. Ontologies are formal structures supporting knowledge sharing and reuse. They can be used to represent explicitly the semantics of structured and semistructured information enabling sophisticated automatic support for acquiring, maintaining, and accessing information. Ontologies provide meaning, organization, taxonomy, agreement, common understanding, vocabulary, and a connection to the 'real world' [Fen04].

John Sowa, in his book, *Knowledge Representation* [Sow00] provides the following definitions:

“The subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called ontology, is a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses a language L for the purpose of talking about D .”

“Ontology: A classification of the types and subtypes of concepts and relations necessary to describe everything in the application domain.”

Another definition which characterizes very well the essence of ontology was given by Grueber: *a formal, explicit specification of a shared conceptualization* [Gru93]. A 'conceptualization' refers to an abstract model of some phenomenon in the world which identifies the relevant concepts of that phenomenon. 'Explicit' means that the type of concepts used and the constraints on their use are explicitly defined. 'Formal' refers to the fact that the ontology should be machine readable. 'Shared' reflects the notion that ontology captures consensual knowledge, that is, it is not restricted to some individual, but accepted by a group [Fen04].

Ontologies link two important parts that make the Web really strong: on one side, ontologies define formal semantics for information; therefore they allow information processing by a computer. On the other side, they define real-world semantics, which makes possible the glue between machine-processable content with meaning for humans.

Ontology is one of the three aspects of knowledge representation, as multidisciplinary subject which applies theories and techniques from logics, ontology and computation. The main problem which ontology poses is that sometimes two entities with different definitions are intended to be the same. Different systems use different names for the same kind of entities, or even same names for different kinds. This is the problematic point in sharing ontologies. Proving that they are actually the same can be tedious, sometimes even impossible [Sow00]. This is one of the aspects treated by XML Topic Maps (see PSIs section, chapter 4) and meant to bring a bit more light in this matter.

3.7 Summary

The previous sections presented different kinds of expressing and structuring knowledge. There are strong associations between them: an ontology describes the main concepts of a specific domain, a taxonomy says more about the relationships between these concepts, while thesauri, indexes and glossaries define in different manner the concepts and their relationships.

XML Topic Maps come also into the light, as it will be shown in the next chapter, and give the possibility to knowledge structures to be expressed as structured link networks, shared and merged. They can be employed to create sophisticated finding aids that allow searching and visual browsing of complex knowledge structures [Sig02].

We have seen in chapter 2 that different businesses and services are registered in a somehow normalized XML database like UDDI and that the way how the knowledge is organized there, is not very suitable for service discovery. Therefore, the domain specific knowledge should be organized in a way that facilitates searching. The structure of the previously mentioned classification systems seems to be well suited to achieve better matching results.

The next step would be to express domain specific knowledge in a descriptive language of the Semantic Web. XML Topic Maps is such a language and the next chapter will give a detailed introduction and show its relation to the Semantic Web as well as why they are a good choice for organizing knowledge, going beyond the traditional methods and also extending it in ways which improve information retrieval.

4 XML Topic Maps

Enormous efforts are spent on extracting the meaning of information and capturing its semantics. Data analysis, manual indexing, cataloguing, classification or thesaurus creation can only be performed reliably by humans, because they require intellectual effort. Unfortunately the results of all that effort are seldom reusable across applications and this leads to tremendous waste, duplication, and redundancy. The Topic Map paradigm addresses this problem. It provides a simple and flexible data model to capture your knowledge, and a standard interchange syntax that makes knowledge accessible across applications. Existing ontologies, document schemas, metadata, taxonomies, glossaries, thesauri and classification schemes, like described in the previous chapter, already carry some level of semantics, which can be mapped automatically into Topic Maps and then successfully implemented [Ont].

Information users should not be constrained to use a single ontology, taxonomy, glossary or other implicit worldview. On the Web, we should exploit different worldviews simultaneously; even they are incompatible with each other [BN01]. Metadata that helps users to find what they need is sometimes too important to limit the possibility of exploiting it to a single proprietary environment. The process searching and finding information should not depend on applications or vendors, such that users are free to consume it in each way they want. The conceptual framework of Topic Maps feeds this need, that is, constitutes itself as a solution for interchanging and handling information produced and maintained by different sources, no matter from which perspective is this viewed.

4.1 History of XML Topic Maps

Originally, there have been markup languages, followed by SGML and SGML Topic Maps. Then followed XML and XML Topic Maps (named XTM). Topic Map interchange syntax was first formalized by ISO standard 13250 (known by the acronym HyTM, for HyTime Topic Maps) in December 1999. After a while, the parties of HyTM formed TopicMaps.Org, with the founding cochairs - Michel Biezunski (of InfoLoom) and Jack Park (of Coolheads Consulting). TopicMaps.Org had as objective the bringing of the Topic Maps paradigm to the Web as XTM.

There are two different Topic Map specifications; one is an ISO standard ISO/IEC 13250:2000 which specifies a standard interchange syntax based on SGML and HyTime, and one is a XML specification (XTM) developed by TopicMaps.Org (which defines an interchange syntax based on XML and Xlink). This thesis deals primarily with the XML specification of Topic Maps, so when mentioning Topic Maps, it will be meant XML Topic Maps.

XML was created because many users felt that SGML was too complicated. There was a need to simplify and limit its features to those that are essential for use in a Web context. XTM was designed with the same motivation: to simplify the ISO Topic Map specification for optimized use on the Web. An XTM Topic Map remains an overlay on information resources as described by ISO 13250. One of the main differences between the ISO Topic Maps specification and XTM is the fact that addressing in XTM is limited to Uniform Resource Indicators (URIs) while addressing in the ISO specification can be expressed with virtually any kind of notation. This restriction recognizes the central role played by the Web as a worldwide network used as a common, well-defined (in contrast to the chaos web) platform on which information can be interchanged [Par02]. This point will be exploited in the case study of chapter 5.

4.2 What are XML Topic Maps

Driving from one place to another without a map could be sometimes challenging, but when the aim is to arrive as quickly as possible, some kind of map is necessary. Similarly, thanks to the Web, we don't suffer from lack of information, but it is problematic to find rapidly, directly accessing the needed information. Therefore what is required is a proper information structuring. Topic Maps is one of the technologies which solve this problem. The case study deals with this issue.

Topic Maps are abstract structures that can encode knowledge and connect the encoded knowledge to relevant information resources. They are organized around *topics*, which represent subject of discourse; *associations*, representing relationships between the subjects; and *occurrences*, which connect the subjects to pertinent information resources. Topic Maps may be represented in many ways: using Topic Map syntaxes in files, inside databases, as internal data structures in running programs, or even mentally as a structuring principle in the minds of humans. All these forms are different ways of representing the same abstract structure [DM05].

Topic Maps are metadata which are not necessary included in the information they describe. They are sort of glue between object of information from distinct sources that make possible that all pieces of information relevant to a certain concept, to be associated one with each other.

The purpose of a Topic Map – based document is to interconnect semantically heterogeneous information. A Topic Map allows readers to navigate topics that can appear in multiple documents. Rather than just being a simple term, a *topic* is a link that contains a title and is pointing to places (called sometime anchors) in the documents where they are occurrences of this topic. A Topic Map is functionally equivalent to multi – document indexes, glossaries and thesauri [PG04].

4.3 Characteristics

4.3.1 TAO of Topic Maps

While it is possible to use Topic Maps for building extremely complicated structures, in the next sections the primary constructs of the Topic Map Model – TAO (Topics, Associations and Occurrences) will be explained.

4.3.1.1 Topics

The best way to think of a Topic is that it is a 'location', a 'container', 'binding point', essentially the place we go to find out 'everything what is knowable' about a 'subject'. A Subject is whatever we may want to talk about.

A Topic is a 'proxy' for a Subject, the object we can store and manipulate in computers. Subjects are things we really cannot put into a computer. We can only find a way to prove the 'identity' of a subject and then talk about it through its proxy, a topic (as representation of the subject).

A Topic is the place, object, whatever, where we collect:

- subject identity (identify the subject this topic is about),
- names for the topic,
- occurrences of the subject which reside 'out there'.

In the newer Topic Map Reference Model [RM03], a topic also keeps track of 'casting' a topic into roles in associations. Right now, in XTM, that does not happen. Instead, in order to know which roles a subject plays (for example, if the subject happens to be the father of somebody, the subject is a 'member' of an association where he plays the role of 'father'), we must go through all the

associations in an XTM document to find those roles. The new Reference Model also lets us attach other properties to the subject, like, for instance, things we could also attach through associations or occurrences, let's say, weight, height, sales price, and whatever [Par04].

So, the 'theoretical' topic should contain:

- subject identity (identify the subject this topic is about),
- names for the topic,
- other properties of the subject,
- casting into roles played in associations by the subject.

When you are an actor, you get cast as somebody or something, into a role. You get cast as the villain, the hero, whatever. The role and the assertion all know what got cast as an actor, that is, the role player [Par04].

The topic map may consist of as many topics as the author wants to put in. How much information she/he is willing to provide, how deep to dig into the given theme is totally up to her/him. The number of topics will determine the size and complexity of the topic map.

The Figure 4 - 1 below shows some topics as example:



Figure 4 – 1: topics

A topic has a *topic type*, or it may have multiple topic types. The topic types are topics as well. Topic types are kind of class – instance relationships, they show the classes in which the topics are grouped.

According to the example from Figure 4-1, Duisburg is a topic of type *city*, Germany of type *country* and DIW (German Institute for Economic Research) of type *institute*. In the Figure 4 – 2, different topic types are represented by different geometrical figures.



Figure 4 – 2: topic types

4.3.1.2 Associations

Topics can participate in associations with one another. Classes of topics, classes of associations, and the roles played in associations are all user-definable, and they are all topics (for example, they are the subjects of topics that represent them). Associations are composed of members - each topic participating in an association plays a specific member role in the association [BN01].

Sometimes it is difficult to find the best associations, that is, to figure out which relationship the topics have to each other. The associations can be like: *lives in*, *written by*, *located in*, or more specific like: *is a scientific organization within*, *it is necessary to satisfy the condition*, etc. A simple example for an association can be: *University of Osnabrueck is located in Lower Saxony* (where *University of Osnabrueck* plays the role of institution and *Lower Saxony* plays the role of state!). In order to avoid confusions, there should be defined *association roles*, otherwise someone could interpret that *Lower Saxony is located in University of Osnabrueck*.

4.3.1.3 Occurrences

A topic can be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are the *occurrences* of the topic. An occurrence could be, for instance, a monograph on a particular topic, or an article about the topic in an encyclopedia, a picture or video depicting the topic, or a simple mention of the topic in some other context. The occurrences are, generally, external to the topic map document itself, although it might be also in it, and they are *pointed at* using, typically, URIs [Pep04].

4.3.2 Name and NameBase

A topic has usually a name, but it can also have no or several names, and each name can take several forms. The most common case is a topic with one name. But having multiple names makes it possible to describe interchangeable networks of knowledge. For instance, each mineral has a scientific name and a common language equivalent. Some terms have different spellings or aliases. Topic Maps do not connect names together, but they connect topics that could have multiple names [Bie02].

Topic Maps use scopes (see section 4.3.3), which are topics, to differentiate names, as for instance the scientific name, maybe Latin, and the common name, maybe English, as scopes on the names given to a topic.

It is quite common for a topic map to have no name. For instance, a formulation like ‘For more details about the concept X, go to ...’ is actually not exploitable, but in Topic Maps terms, this reference to another location can be rendered as two different occurrences of the same topic [Bie02].

4.3.3 Scope and Namespace

According to [XTM01] scope is said to specify the limit of the validity of a topic's characteristic (association, name or occurrence). It establishes the context in which a name or an occurrence is assigned to a given topic, and the context in which topics are related to each other through associations. Every characteristic has a scope, which may be specified either explicitly, as a set of topics, or implicitly, in which case it is known as *unconstrained scope*.

For example, define a scope consisting of the subject "English" using a published subject [XTM01]:

```
<scope>
  <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
</scope>
```

A Namespace is a set of base names of one or more topics, each of which is unique and all of which are the names of their respective topics within a single, common scope [Bie02].

4.3.4 Published Subjects and Published Subject Indicators

Published Subjects provide a mechanism whereby computers and also humans, in interaction with computers, can know when they are talking about the same thing. They establish the identity of subjects of discourse. The importance of establishing identity is doubtless: without it, there can be no communication between humans and no operability between applications [PG04]. XTM gives one way, called Published Subject Indicator (PSI), which is typically an URI or URL and, by social agreement, establishes the identity of some subject.

The usual way of identifying subjects in today's networked environment is through addressing, more precisely, through URIs. This works quite well when a subject has an unambiguous address, namely a network-retrievable information resource. The address of a subject which is an information resource is simply called *subject address*. What about the subjects which are not information resources? The identity of these kinds of subjects can be established *indirectly*. Topic Maps provides an approach for doing this through *subject indicators*; a subject indicator is different from the subject that it indicates. It can be a textual definition, name or description; it can be audio, video or some other type. From the perspective of an application, if the identity of two topics is established by subject indicators that have the same address, they should be regarded as representing the same subject. The address of a subject indicator is called *subject identifier*. *Subject indicator* and *subject identifier* are two faces of the same identification mechanism; the former being for humans, while the latter for applications [Oas03].

Let us summarize. The *subject identifiers* (URIs) identify the subject the topic is about. They point to resources which describe the subject to a human. The resources are *subject indicators*.

There are three ways of indicating a subject in Topic Maps:

- pointing via a `<topicRef>` element to a `<topic>` element that *shares* the same subject,
- pointing via a `<subjectIndicatorRef>` element to a resource that *indicates* the subject
- pointing via a `<resourceRef>` element to a resource that *is* the subject.

The subject indicated by a subject indicator may be either non-addressable or addressable.

4.4 Merging XML Topic Maps

<mergeMap> is a very important element in XTM; it merges two or more topic maps into one topic map. Either all topics with the same name in the same scope can be merged (the so-called *name-based merge*), or all topics with the same subject identity are merged (a *subject-based merge*).

The process of merging ensures that whenever two topics are known to represent the same subject, they are merged. A unique topic characteristic is one where the value of the characteristic effectively identifies the subject of the topic. That means that if two different topics have the same value for a unique characteristic, they also represent the same subject, and so must be merged [DM05].

Reification is the act of creating a topic. When anything is reified, it becomes the subject of the topic thus created; to reify something is to create a topic of which that thing is the subject [XTM01].

But why should one merge two topic maps? When two topic maps are merged into a single one, that topic has all the topic characteristics of both topics – base names, occurrences, roles played in the associations, hence the benefit of it. If one choose to give topics that others think have different subjects the same name, it makes sense to merge them, as well as if others think that topics to which you give different names have the same subjects [Hun02].

4.5 Creating an XML Topic Map

For the simple taxonomy drawn in Figure 4-3, we will create an XML Topic Map, with the purpose of showing step by step how an XTM can be born. One

should be aware that XTM and Topic Maps are different things [Gar05]: Topic Maps are the concepts and the model, while XTM is just the syntax used to transport model instances around.

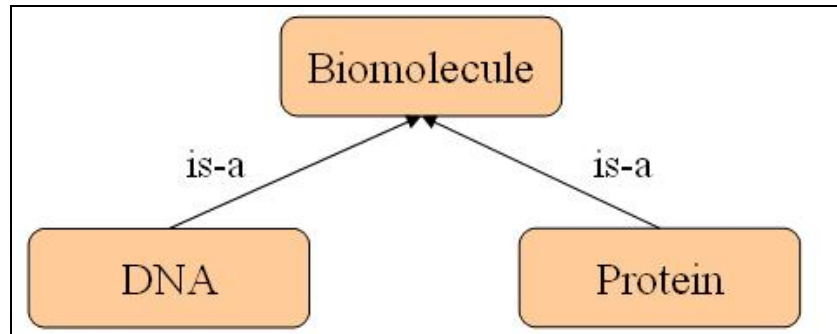


Figure 4 – 3: Biological taxonomy

We will introduce some of the XTM elements: `<topic>`, `<baseName>`, `<scope>`, `<baseNameString>`, `<occurrence>`.

Let us create topic elements and give them IDs for identifying them:

```
<topic id="Biomolecule">, <topic id="DNA"> and
<topic id="Protein">.
```

Since we want to talk about *Biomolecule*, *DNA*, *Protein*, let us type a base name for our topics. We will pack them in a `<baseNameString>` like that:

```
<topic id="Biomolecule">
  <baseName>
    <baseNameString>Biomolecule</baseNameString>
  </baseName>
</topic>
```

```
<topic id="DNA">
  <baseName>
    <baseNameString>DNA</baseNameString>
  </baseName>
</topic>
```

```
<topic id="Protein">
```

```
<baseName>
  <baseNameString>Protein</baseNameString>
</baseName>
</topic>
```

Now we supply a little more information for what we are talking about, by adding an image of a DNA. In the same way can be proceeding for the other elements of the taxonomy.

```
<topic id="DNA" >
  <baseName>
    <baseNameString>DNA</baseNameString>
  </baseName>
  <occurrence>
    <resourceRef xlink:href="DNA.gif" />
  </occurrence>
</topic>
```

The *xlink:href* attribute can be thought as the *href* attribute of an HTML element. It points to a resource, in our case to the GIF file that is the occurrence of the DNA topic. The *xlink:* prefix shows that the attribute conforms to the XLink World Wide Web Consortium specification.

Now we will introduce some other elements: `<resourceRef>`, `<subjectIdentity>`, `<subjectIndicatorRef>` and `<topicRef>`.

We give now to the "DNA" topic an identity that both machines and humans can understand, namely the URI:

<http://www.rothamsted.bbsrc.ac.uk/notebook/courses/guide/images/dna.gif>.

Thus, the subject of our topic can be declared in the following way:

```
<topic id="DNA" >
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
"http://www.rothamsted.bbsrc.ac.uk/notebook/courses/
guide/images/dna.gif" />
  </subjectIdentity>
  <baseName>
    <baseNameString>DNA</baseNameString>
  </baseName>
```



```
<occurrence>
  <resourceRef xlink:href="DNA.gif"/>
</occurrence>
</topic>
```

Let us see how the element `<scope>` in our example looks like and to give to our topic a second name in German:

```
<topic id="DNA">
  <baseName>
    <scope>
      <topicRef xlink:href="#EN"/>
    </scope>
    <baseNameString>DNA</baseNameString>
  </baseName>
  <baseName>
    <scope>
      <topicRef xlink:href="#DE"/>
    </scope>
    <baseNameString>DNS</baseNameString>
  </baseName>
  .
  .
  .
</topic>
```

Let us now set up the `<topic>` elements to point at and use PSIs to express their subject identities:

```
<topic id="EN">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
  </subjectIdentity>
</topic>
<topic id="DE">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/language.xtm#de"/>
  </subjectIdentity>
</topic>
```

Let us now introduce the following elements: `<associations>`, `<member>`, `<roleSpec>` and `<instanceOf>`.

An association is a relationship between one or more topics, each of which plays a role as a member of that association. The `<roleSpec>` element specifies the role played by a member in an association. The `<instanceOf>` element specifies the class to which its parents belongs, via a `<topicRef>` or `<subjectIndicatorRef>` child element [XTM01].

```
<association>
  <instanceOf>
    <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class-
instance"/>
  </instanceOf>
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#instance"/>
    </roleSpec>
    <topicRef xlink:href="#DNA"/>
  </member>
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class"/>
    </roleSpec>
    <topicRef xlink:href="#Biomolecule"/>
  </member>
</association>

<association>
  <instanceOf>
    <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class-
instance"/>
  </instanceOf>
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#instance"/>
    </roleSpec>
    <topicRef xlink:href="#Protein"/>
  </member>
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class"/>
    </roleSpec>
    <topicRef xlink:href="#Biomolecule"/>
  </member>
</association>
```

We have presented in this section a possible XML Topic Maps construction, a very simple one, which was based on the taxonomy given at the very beginning of the chapter. Of course, we could add much more complicate and meaningful attributes and create correspondent topics and associations, but the point here was just to get familiar with the XTM constructs and with the way of creating them. The whole XTM document can be found in Appendix A.

4.6 Some tools for XML Topic Maps

The creation of knowledge representations is not thought to be done by hand coding, but with helpful tools. Therefore this section will give a small overview on software and tools that exist and that can be used for creating, navigating, in general working with XML Topic Maps. We concentrate here on freely available tools. The following table gives some information about such tools.

Topic Map Tool	Description	available at
TM4J	Topic map processing engine implemented entirely in Java and providing a pure Java API; among other facilities, supports the import and export of XTM syntax.	http://tm4j.org/
Perl XTM	Enables to read and write XML Topic Maps and to manipulate and query Topic Maps via an API.	http://search.cpan.org/dist/XTM/
Topic Map Designer	Tool for creating and viewing Topic Maps (tool in development).	http://www.topicmap-design.com/
TMTab	A tab-widget plug-in for Protégé (a free, open source ontology editor and knowledge-base framework) which enables the use of Protégé as a topic map editing tool. The current plug-ins functionality is limited, because it currently supports only the	http://www.techquila.com/tmtab/index.html

	export of XTM files from an ontology developed in Protégé, which is based on the “topic map ontology” distributed with the plug-in. Doesn’t support the import of already existing ontologies.	
Ontopia Omnigator	Allows browsing XML Topic Maps in a convenient web interface without programming and configuration.	http://www.ontopia.net/download/freedownload.html
Hyper Graph	An open source tool for visualizing and navigating Topic Maps, where each topic is represented as a node in the graph. Program is in test phase.	http://www.ontopia.net/download/freedownload.html
SemanText	A prototype application developed to prove how the topic map standard can be used to represent semantic networks.	http://www.semantext.com

However, there are quite some tools freely available, but they are still not mature and represent just prototypical implementations.

For instance, Protégé is one of the most popular tools for ontology editing. Currently it provides e.g. the possibility to save ontologies in OWL or RDF format. The TMTab tab-widget plug-in, which can be downloaded additionally for use with Protégé, supports the export of XML Topic Maps. But it has some limitations. It is not possible to export an already existing ontology to XTM format. TMTab acts as an editing tool, but it merely supports the export of ontologies that are based on the predefined “topic map ontology” distributed with the plug-in. Even Kal Ahmed, the developer of TMTab (and also of TM4J) is of the opinion that there are no really good editing tools for topic maps right now (mail on January 23, 2005).

Writing XTM by hand is very hard as the example taxonomy from Figure 4.3 showed. Therefore it is quite recommendable to use the LTM (Linear Topic Map) syntax. Leading personalities of the Topic Map community, among them

Kal Ahmed or Lars Marius Garshol from Ontopia, suggested LTM as the easiest way to write topic maps (mail on January 23, 2005).

As far as the tools are still not mature and writing XTM by hand is hard, the prototypical ontology will be developed with the help of the LTM notation. Such a LTM file can be loaded and viewed e.g. with the freely available tool Ontopia Omnigator. This tool even provides the possibility to export an ontology created in LTM to an XTM file. The next section will therefore give an introduction to Linear Topic Maps and their syntax.

4.7 LTM as an alternative for editing and creating Topic Maps

The linear topic map notation is a simple textual format for topic maps. Just like the XML interchange format, it represents the constructs in the topic map standard as text, but unlike the XML format it is compact and simple. The notation can be written in any text editor and processed by topic map software that supports it, or converted into the XML format supported by such software [LTM02]. One can use, for example, either TM4Web or Omnigator to view LTM files directly. At the end of this section there will be given a small description of the Omnigator.

In LTM, the XML-based topic map interchange format is defined in a comprehensible way for humans, as well as for developing software for it. It was developed by Ontopia – a commercial company, the world’s leading Topic Maps application development platform, whose initiator was Steve Peppers; the notation was later refined by Lars Marius Garshol.

In the following, we will discuss briefly how the topic maps’ elements in LTM notation look like. We will practice on the previously chosen example, the biological classification. As consequence, the simplicity of it will elect them as a good choice for working manually with topic maps, although some disadvantages exist.

Defining topics – the primary constructs in topic maps, is done in LTM by simply writing its ID in square brackets, i.e.

```
[biomolecule]
```

If wanted, a baseName can also be provided:

```
[biomolecule      = "Biomolecule"]
```

In order to display the type of the topic, it will be done like that:

```
[dna : biomolecule = "DNA"]
```

In case the topic has more than just one type, the multiple type IDs will be listed one by one, separated by colons or white spaces.

Subject indicators for topics can also be provided by giving the URL of the subject indicator, quoted and preceded by a “@” character:

```
[dna : biomolecule = "biomolecule";  
  @"http://www.rothamsted.bbbsrc.ac.uk/notebook/  
  courses/guide/images/dna.gif"]
```

For topics which represent network-retrievable resources, the reference for indicating the identity of the subject can be done directly, and in this case it is used ‘%’ character, followed by the quoted URL of the resource.

LTM support also scoping names, i.e.:

```
[dna : biomolecule = "DNA" = "DNS" / de]
```

The associations look in LTM like that:

```
is-a(dna, biomolecule),
```

while the occurrences:

```
{dna, illustration,  
"http://www.rothamsted.bbbsrc.ac.uk/notebook/courses/  
guide/images/dna.gif" }
```

LTM provide also support for reifying topic maps themselves, specifying a base URI for the topic map, merging in external topic maps, and specifying scope on occurrences and associations. But this kind of details is out of the scope of this section.

Comments can also be added, in order to explain step by step the significance of the following piece of codes.

Summarizing, the LTM topic map for the mentioned example will look like:

```
#TOPICMAP bio-tm  
#INCLUDE "dc.ltm"  
  
/* topic types */  
  
[biomolecule = "Biomolecule"; "compound of living  
organisms"]  
[dna = "DNA"; "deoxyribonucleic acid"]  
[protein = "Protein"; "enzyme"]  
  
/* association types */  
  
[is-a = "Is a"]  
  
/* occurrence types */  
  
[illustration = "Illustration"]  
  
/* topics, associations, occurrences */  
  
[biomolecule= "Biomolecule"]  
  
[dna : biomolecule = "DNA" = "DNS" / de]  
  
[protein : biomolecule = "Protein"]
```

```
is-a(dna, biomolecule)
is-a(protein,biomolecule)

{dna, illustration,
  "http://www.rothamsted.bbsrc.ac.uk/notebook/
  courses/guide/images/dna.gif"}
```

Looking at the entire XTM document from the appendix A and at the above LTM one, it is more than obvious which one is easier, therefore preferred to work with.

But there is an important aspect which should be taken into consideration. Although the LTM syntax supplies a very convenient way for maintaining the Topic Maps, one should be aware of the fact that the only standardized forms for interchangeable Topic Maps remain the XTM 1.0 and HyTM syntaxes. There is a Technical Report of the Linear Topic Map Notation, published by Ontopia, but not a formal specification from any recognized standards body. The specification may be taken up by some standards body at some point, but nothing suggests that this will happen soon [LTM02].

As mentioned at the beginning of this section, we introduce now the Ontopia Omnigator. The Omnigator is a technology showcase and teaching aid designed to reveal what the Topic Maps can do. It is used also as topic map debugger and prototyping tool [Omn05]. An interesting feature of the Omnigator is that gives the possibility to load and navigate Topic Maps manually (see Figure 4-4), regardless of its format – XTM, HyTM, LTM, or even RDF.

omnigator eight
The free Seamless Knowledge browser from **Ontopia**. Powered by the **Ontopia Knowledge Suite**.

Welcome | Manage | Customise | Filter | Export | Merge | Statistics | Query | Reload | Not indexed | Feedback | No schema

Index Page

Topic Map Overview

- **Ontology**
- Master Index
- Index of Individuals
- Index of Themes
- Unnamed Topics (2)

Topic Types (3)

- Biomolecule
- Format
- Type

Association Types (1)

- Is a

Association Role Types (1)

- Biomolecule

Occurrence Types (1)

- Illustration

Figure 4 – 4: Navigating biological taxonomy

But the Omnigator does much more than browsing topic maps: it has the ability to merge topic maps, searches through them as Google does, performs semantic validation and exports to a range of syntaxes. Regarding the export, a practical thing is that Omnigator allows exporting documents from LTM in XTM syntax, as shown in Figure 4-5. Also the import of any conforming topic map format is possible.

omnigator eight
The free Seamless Knowledge browser from **Ontopia**. Powered by the **Ontopia Knowledge Suite**.

Welcome | Topic Map

Export

Using this plug-in you can export the current topic map as an XML document, in HyTM, XTM, or RDF syntax. You can choose to either download it or view it in your browser.

In addition this plug-in offers the possibility to generate a topic map schema from the information found in the current topic map. The syntax used for the schema representation is set to Ontopia's Schema Language (OSL).

Syntax	Action
<input checked="" type="radio"/> XTM 1.0 syntax	<input checked="" type="radio"/> Download
<input type="radio"/> HyTM syntax	Save as: <input type="text" value="bio.ltm.xml"/>
<input type="radio"/> RDF/XML syntax	<input type="radio"/> View
<input type="radio"/> CXTM syntax	<input type="button" value="Export"/>
<input checked="" type="radio"/> OSL syntax	<input checked="" type="radio"/> Download
	Save as: <input type="text" value="bio.ltm.osl"/>
	<input type="radio"/> View
	<input type="button" value="Generate schema"/>

(C) Copyright 2001–2004 **Ontopia**. All rights reserved. **Download** the latest version when this one expires!

Figure 4 – 5: Exporting a topic map document to different syntaxes

4.8 Summary

This chapter provided detailed information about the concept of XML Topic Maps. It was described what XML Topic Maps are and how they have to be developed. An example showed how a simple XML Topic Map has to be created manually. Further it was seen that there are currently no good tools freely available for creating topic maps. As an alternative to those immature tools the LTM notation was presented. It provides a simple syntax and there exist tools that allow an export of LTM files to XTM files. We will take advantage of this in the case study.

5 Case Study

5.1 Introduction to the case study

The previous chapters gave a detailed introduction from a theoretical point of view, where the involved technologies have been described. In order to show the benefits of combining technologies from the area of the Semantic Web with technologies from the Web Services area, the case study will tackle the problem from a practical point of view.

Consider following **use case scenario**: A biologist is searching for a specific (previously unknown to him) Web Service out of the biological domain. This case study will show how the discovery problems described in section 1.2 and 2.2.3 could be solved through an XTM solution. This leads to an improvement of the search process. Such a solution is new, and therefore, to my knowledge, there is no implementation so far.

We follow a predefined methodology. This is the reason why the creation of an ontology has to be well planned in order to avoid problems during the development and to ensure that the project is a success.

“There is no one correct way to model a domain – there are always viable alternatives.” One such alternative, for instance, was developed at the Stanford University [NM00]. It describes a possible way to develop ontologies with the Protégé software tool.

However, such a methodology to model a domain represents a kind of project management with different milestones. Section 5.4, with the prototypical implementation, will also follow a simple methodology. But before we switch to the implementation section, some more knowledge is needed. As far as the case study considers the biological domain, the next section will be concerned

with biological ontologies. This is very important, because generally it is envisioned to reuse ontologies. Section 5.3 is also very important, because it will show how the solution can be reached in technical terms, i.e. how to link an ontology to UDDI. In section 5.5, after the prototypical example, it will be shown how to navigate/query the developed topic map solution. A small comparison to alternative techniques follows, before the chapter concludes with a summary.

5.2 Biological ontologies

Ontologies are getting more and more weight as a central component of many applications. In bioinformatics (the discipline of applying computing to molecular biology), ontologies gained power in the past years, motivated by the need to consistently access, describe, share and control the over 500 databases in routine use. The complex biological data stored in bioinformatics databases requires often the addition of knowledge to specify and constrain the values held in those databases. Moreover, many biologists are working by applying prior knowledge to an unknown entity. Ontologies are generally viewed as a promising mechanism for expressing and sharing the molecular biology community's challenging rich, dynamic and complex knowledge. While the amount and complexity of that knowledge increases, the need to exploit it efficiently and effectively grows [GO04]. Note that ontologies play an important role also for other new fields (nanotechnology, etc.); in our case bioinformatics has been chosen as 'universe of discourse'.

Biologists waste currently a lot of time and effort in searching for all of the available information about each small area of research. One of the biggest names in the bio-ontology world is Robert Stevens, whose interest area is the building and use of ontologies and the reconciliation of semantic heterogeneity within bioinformatics resources. In [SGB02], Robert Stevens et. al. are giving an introduction to this domain and a survey of current existing bio-ontologies. These bio-ontologies are very different and specific to their intended use. The

following paragraphs provide a condensed overview on some of the most important ontologies in this domain.

The **Gene Ontology** (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The GO collaborators are developing three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated *biological processes*, *cellular components* and *molecular functions* in a species-independent manner. The use of GO terms by several collaborating databases facilitates uniform queries across them; the controlled vocabularies are structured in such a way, that one can query them. GO terms (over 15.000) are organized in structures called *directed acyclic graphs*, differing from hierarchies in that a child (more specialized term) can have many parents (less specialized terms) [GO04].

The GO is becoming the de facto standard for describing gene products in terms of their molecular functions, biological processes in which they participate, and the cellular locations in which they are active. Relying upon a DAML+OIL version of GO, including mined GO-term-to-gene-product-type and GO-term-to-GO-term associations, and the FaCT reasoner (a classifier of description-logic-based ontologies), GOAT (The Gene Ontology Annotation Tool) aims to guide the user in the annotation of gene products with GO terms by displaying those field values that are appropriate based on previously entered terms. This will result in annotations of a higher quality, which in turn will facilitate biomedical e-Science [BMWS].

GOAT [GOAT04] is closely related to another project at the University of Manchester named GONG (Gene Ontology Next Generation). The goal of GONG is to convert the present GO into a description-logic-based ontology (specifically, in DAML+OIL) and then to further enrich it with formally represented biological knowledge.

TAMBIS (Transparent Access to Multiple Biological Information Sources) [TA98] is a joint research project between the School of Biological Science and the Information Management Group, in the University of Manchester in the UK. TAMBIS aims to aid researchers in biological science by providing a single access point for biological information resource. The TAMBIS Ontology (TaO) is an ontology of a molecular biological and bioinformatics concepts. The TaO is mostly used to:

- describe the metadata of the underlying data sources, representing an over-arching universal schema;
- form queries expressed in the modelling language;
- mediate between the various sources to translate the mediator's model to the sources' models.

It has as main components: an ontology of biological terms in GRAIL (GALEN Representation and Integration Language) and now in DAML+OIL, a knowledge-driven query formulation interface, a sources and services model linking the biological ontology with the source schemas and a query transformation rewriting process [TA98].

These are just some of the initiatives and projects in the bioinformatics direction.

5.3 Linking the XML Topic Map to UDDI

It is clear that a knowledge representation system has to be linked somehow to the UDDI registry in order to be useful. The following subsections are dealing with this issue. In most of these conceivable options the XML Topic Map acts as a topic/semantic layer over the UDDI registry. The first two sections consider that each service on the net is a resource with metadata. Section 5.3.3 on the other hand deals with the topic in a different manner, because it

represents a solution, where the knowledge representation expressed in XTM acts as a kind of registry itself, and therefore replaces the role of UDDI, like described in [Zar04].

5.3.1 Ontology to UDDI

It was mentioned earlier that the semantic problem with respect to UDDI was also identified by the UDDI community. This and the following subsection consider and describe some approaches, treated in a discussion paper by the UDDI Technical Committee [Vos05].

Figure 5-1 shows the first possibility of how an ontology could be linked to an UDDI entity. An ontology element is referencing an UDDI entity. The uniqueness of the UUID key (Universal Unique Identifier) enables the linkage between a knowledge representation and an UDDI entity.

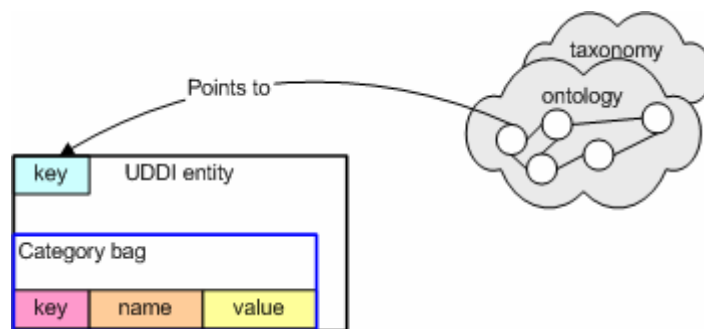


Figure 5-1: Ontology points to UDDI entity [Vos05]

5.3.2 UDDI to Ontology

This section shows how a vice-versa connection from the UDDI entity to an ontology looks like. In this solution the service requestor finds an UDDI entity

and wants to know more about it. A category bag is here pointing to an ontology. This solution implies that the service provider has to link its UDDI entity to the ontology. The previous idea could be used also by somebody else than the service provider. That's why we will follow only the previous and the coming solution in our prototypical implementation. Figure 5-2 shows visually how this approach looks like.

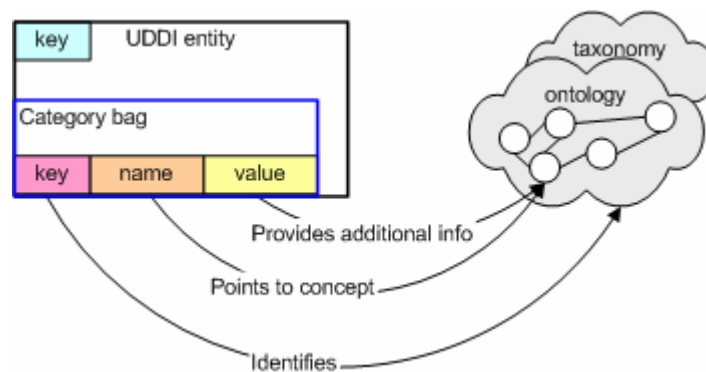


Figure 5-2: UDDI entity points to an ontology [Vos05]

5.3.3 Totally without UDDI

The previous sections showed just some possibilities how to link an ontology to UDDI, more precisely, to an UDDI entity. This would extend the current workflow (Figure 2-1), where the service requestor searches the registry and gets the location of the WSDL file from there, because the requestor would first contact the external ontology, which would refer her/him to the UDDI entity of desire. The title of this thesis originates from this idea.

Following solution is also conceivable. Here the ontology itself would act as a kind of registry and point either to the location of the WSDL file or to the location of the service itself [Zar04], preferably to both.

The new point in this solution is that UDDI is the registry for all kind of services, no matter of the domain and this proposal represents just a kind of *domain-specific solution* represented in XTM. As far as XTM comes from the area of the Semantic Web, this solution offers real semantic search, as it will be described in section 5.5. Additionally, we are dealing here with a specific domain, in our case bioinformatics, and therefore we would not be able to find a service (or resource in general) from another domain like e.g. economy or mathematics.

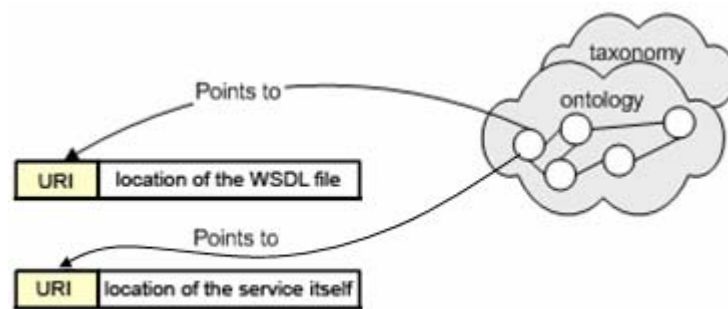


Figure 5-3: Ontology points to the WSDL- location or directly to the service

All three solutions have a common weak point. In case a service provider would change something, e.g. the location of the WSDL file, we would have a dead link. The result is, that the work is not done after developing an ontology, because the knowledge representation has to be maintained in order to ensure a trustful solution and service.

5.4 Prototypical implementation

Whenever something has to be developed or implemented, it is advisable to follow some previously defined steps in order to ensure the success of a project. Section 5.1 already described the importance of such a project management and that there are many different ways for doing it.

Before the development of the topic map started, a step-by-step methodology was defined. These step-by-step instructions turned out to be successful with respect to the development of topic maps. Our methodology consists of following seven steps:

- **Step 1:** Defining the goal
- **Step 2:** Considering additional possible prerequisites
- **Step 3:** Reusing existing ontologies or defining new ones
- **Step 4:** Setting up relevant topics
- **Step 5:** Assigning information resources
- **Step 6:** Setting up associations between the topics
- **Step 7:** Testing the topic map

The following subsections tackle these steps more detailed and describe what actions and questions have been considered during each step.

5.4.1 Step 1: Defining the goal

The first step should include thoughts about the scope of the intended work. The primary aim, with respect to this work, is to improve the discovery process of Web Services in the area of bioinformatics. Therefore, the main consumers of such an ontology will presumably be biologists seeking for previously unknown resources (mainly in form of Web Services) on the internet.

Additionally to the development of an ontology, it has to be shown how this ontology helps achieving the goal, i.e. to show how the ontology can be used through adequate navigation for locating specific resources out of a certain domain. Section 5.5 will show detailed how the topic map can be navigated.

5.4.1 Step 2: Considering additional prerequisites

The second step in this methodology is also very important, because here it should be decided how the knowledge base has to be developed. The prototypical topic map will be developed manually in a simple text editor with the help of the LTM notation for reasons earlier described in the thesis.

5.4.1 Step 3: Reusing existing ontologies or defining new ones

The question whether to reuse existing ontologies or to define new ones has to be answered at this point. It should not be underestimated that the development of a new knowledge representation requires the implication of domain specialists and is quite time-consuming.

Ontologies pretend to represent the knowledge out of a specific domain and therefore envision to be reused. The Gene Ontology, for instance, is a widely used ontology and therefore a trustful source to be reused. Despite of the terminology, the Gene Ontology is not an ontology, but a structured controlled vocabulary. This structure is made up of “nodes” and “edges”, which correspond in topic maps to “topics”, respectively “associations”.

The terminology of the Gene Ontology is used to annotate gene products with respect to three attributes: the *molecular functions* of these products, the higher-level *biological processes* in which they are involved and the *cellular components* where they can be found. These three attributes build up three hierarchies, which are independent of each other, that mean that no links between terms from separate hierarchies exist. In the context of this thesis we will concentrate on the last attribute, namely *cellular component* and out of it the subset *virion*.

5.4.1 Step 4: Setting up relevant topics

During this step we have to set up the topics for our topic map. The Gene Ontology consists of human-readable terms. Each node in the graph is a natural-language term, and for almost all terms, a natural-language definition is given.

It was mentioned earlier that a “node” represents a “topic”. So it is easy to set up the relevant topics and this is one of the main advantages of reusing already existing and well established ontologies. In section 4.7 it was exemplified how topics in LTM are created and look like.

5.4.1 Step 5: Assigning information resources

Section 5.3.1 showed how an ontology could be linked to UDDI. An entry in the UDDI registry represents an information resource. It was also mentioned that each UDDI entity is uniquely identified through a key (UUID). As a result it is not difficult to assign such an information resource.

But this prototypical implementation will not refer only to UDDI entities. Section 5.3.3 described that a topic can refer also directly to the description of a service. Both sections will be considered for the implementation and additionally other information resources will be prototypically assigned. Such resources could for instance be figures, download-files from an ftp-server, links to relevant scientific publications, etc.

Therefore the result can be viewed as a “complete” **knowledge base** for biologists. This knowledge base has a much bigger functionality than the title of this thesis promises, because it does not refer only to UDDI and Web Services, but to resources in general. Note that in the prototypical implementation fictitious locations of information resources were used. This can be achieved through different kinds of “occurrence types”. Figure 5-4

shows what kind of resources can exemplarily be assigned to a topic of an ontology expressed in XTM.

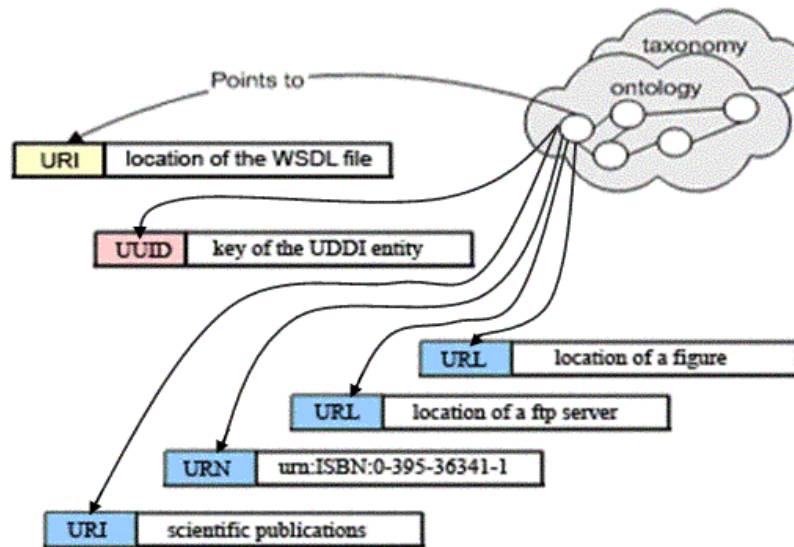


Figure 5-4: Ontology as a complete knowledge base pointing to different information resources in one domain

Remark that the suggested solution becomes an alternative to the UDDI in this way.

5.4.1 Step 6: Setting up associations between the topics

The purpose of this step is to set up associations between the topics developed in step 4. Each of the three hierarchies of the Gene Ontology is constructed from a mixture of taxonomy and partonomy. Therefore each edge in the graph is either a “is-a” or a “part-of” relation.

The facilities offered by LTM permit us to represent the above mentioned relations in a gentle way. For instance, in the Gene Ontology the edge between the terms “viral capsid” and “virion” is a “part-of”-relation. This is represented in LTM like:

```
part-of{viral_capsid, virion}
```

5.4.1 Step 7: Testing the topic map

This is the last step in the development of the topic map. As far as our LTM – document was hand-coded, it is very possible to have some typing errors. In order to prove whether the information contained in our topic map is consistent and meaningful, we used the Ontopia Omnigator for debugging our topic map. In case there are errors, the Omnigator does not load the topic map, but it gives valuable hints regarding the error and its location in the LTM document.

After the LTM was loaded successfully, the export functionality was applied and the topic map was exported to the XTM 1.0 syntax. The complete LTM document of the prototypical topic map can be found in Appendix B. For space reasons the XTM document is not attached to the thesis, but added on a CD coming along with the thesis.

The new “Welcome page” contains now our cell topic map (in both: LTM and XTM), like shown in Figure 5-5. At this point we can say that we have developed successfully a topic map, because it was tested and can now be navigated and used for our purposes.

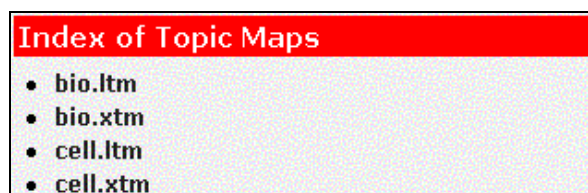


Figure 5-5: Index of topic maps in different formats

5.5 Navigating the Topic Map

After the successful development it will be shown in this section how our topic map can be navigated. Omnigator offers several ways in doing so:

- *Link navigating* the Omnigator HTML pages
- *Visual navigation* through the Vizigator
- *tolog* – a language for querying topic maps.

5.5.1 Link Navigation

Omnigator uses a simple client-server architecture based on a standard http protocol. The application on the server side runs on a Tomcat web server. It reads and writes topic maps and generates HTML pages on the fly. On the client, a standard web browser receives these HTML pages and displays a view of some portion of the topic map. This view is rich in links, built from the data structures that constitute the topic map. Each time the user clicks on a link, a request is sent to the server application, resulting in a new set of information extracted from the topic map [Ont04].

Omnigator visualises a topic map as a structured list of links. By clicking these links a user is enabled to navigate around. In Figure 5-6 the link for the GO term *Viral genome* was clicked and all information on that topic is displayed.

Viral genome

Names (1)

- **Viral genome**

Associations (6)

- **Is a**
 - DNA viral genome
 - Non-segmented viral genome
 - Provirus
 - RNA viral genome
 - Segmented viral genome
- **Part of**
 - Virion

External Occurrences (3)

- **Description**
 - http://psi.virion.org/viral_genome.wsdl
- **Registry**
 - [uuid:w94nfo18-vjr6-v8s0-fibe-bsu62k5n39bp](http://psi.virion.org/viral_genome.wsdl)
- **Service**
 - http://psi.virion.org/viral_genome.pl

Figure 5-6: Example of navigating Omnigator HTML pages

On the right side we can find links to external occurrences, in this case to the location of the WSDL file, to the unique identifier key of an UDDI entity and to the access point of a Web Service concerned with this topic. As it was mentioned earlier, it is also possible to have other information resources like figures, etc.

Remember that the Gene Ontology has the structure of a directed acyclic graph constructed from a mixture of taxonomy and paronomy. We can navigate through this graph by simply clicking the links in the association box on the left side of Figure 5-6. There we can see that *Viral genome* is involved in the “is-a” and “part-of” association to other terms in the graph.

The example shows further that only terms laying at the first level of depth can directly be accessed. The topic map is not displayed as a whole. In case we want to reach a term laying on a deeper level in the graph, we would reach there through clicking from the initial, over the intermediate(s), to the desired level.

5.5.2 Visual Navigation

“Topic Maps can be seen as a network of topics, so network and graph visualization techniques are interesting to the topic map community” [Gra02].

An eye-catching feature in Omnigator Eight is the Vizigator, which produces a graphic visualization of our topic map. The Vizigator is accessed via the Vizigate plug-in and provides the user an additional method of topic map navigation through graphical browsing. The Vizigator enables the user to have a more extensive view on the topic map, because it is possible to choose and access the desired level of depth. In the top left corner of Figure 5-7 we can fill in the box the desired number of levels to be displayed (in our case - 3). Small red boxes with numbers indicate further depths in the representation. In the same figure the two relationships/associations can be distinguished through

different colours; pink edges stand for the “part-of” and purple edges stand for the “is-a” associations. The start topic is always displayed in yellow and represents the topic where the user usually starts her/his manual browsing from. But the Vizigator offers additionally an alternative way to find topics, the search box. All or part of a topic name has to be filled in the search box and the hit will blink in green. For instance, we searched for the GO-term *DNA viral genome*, and the result is like shown in the following figure:

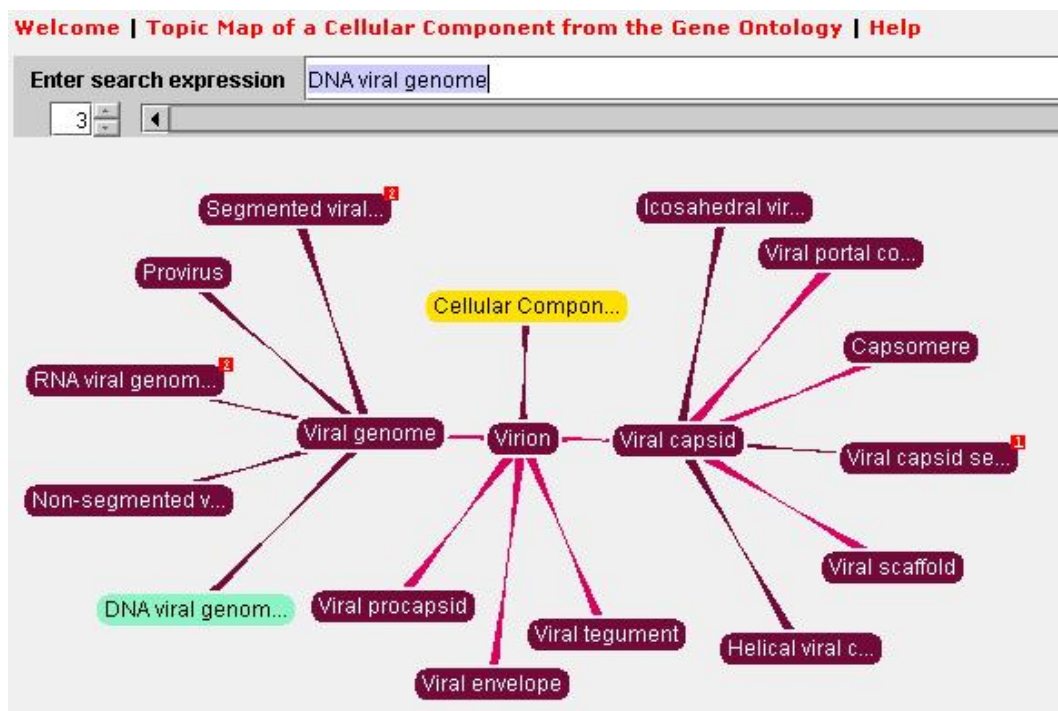


Figure 5-7: Graphically navigating the Cellular Component of GO

Now, by right-clicking on the result (or the topic of desire), we are getting to the details about the external occurrences (Figure 5-8).

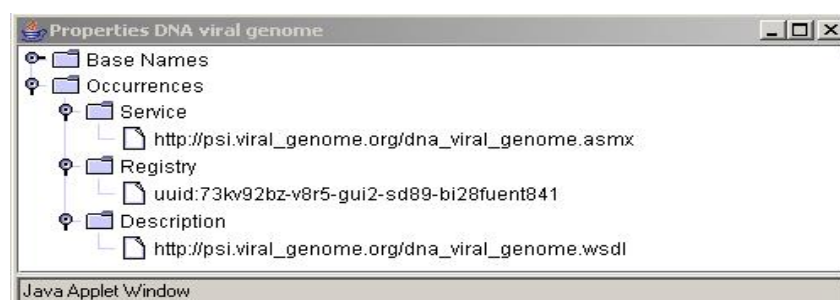


Figure 5-8: Properties with occurrences for a specific topic

5.5.3 tolog – a topic map query language

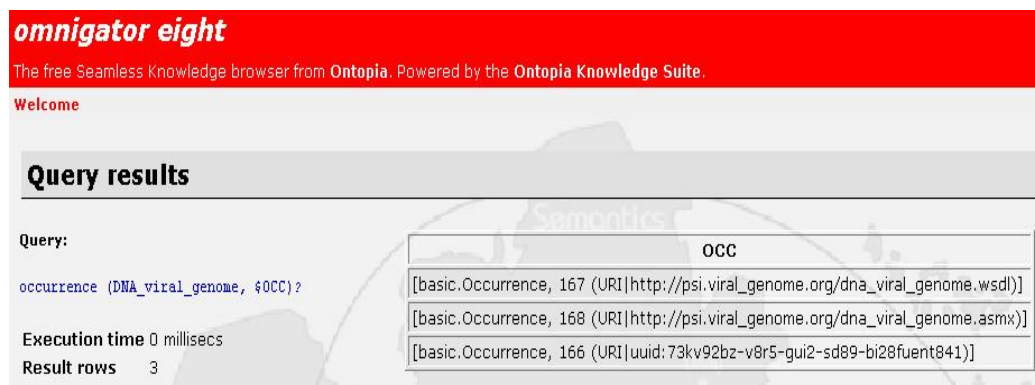
Sections 5.5.1 and 5.5.2 showed two ways of navigating a topic map. This section shows briefly a more “elegant” way for doing it.

tolog is a language for querying topic maps, developed by Ontopia. It is inspired by *Datalog* (a subset of *Prolog*) and *SQL*. Using *tolog* one can query a topic map in much the same way as a relational database can be queried with *SQL*. “*tolog* is a logic-based query language, which means that the basic operation is for the user to ask tolog in which cases a certain assertion holds true, and tolog will then respond with all the sets of values that make the assertion true” [Tol04].

Coming back to the example of last section, where we were interested in external information sources (occurrences) about the GO-term DNA viral genome, a tolog command would look like:

```
occurrence (DNA_viral_genome, $OCC)?
```

The result of this tolog – query is shown in Figure 5-9 and is the same as in the previous section. Detailed information about tolog can be found at [Tol04].



omnigator eight
The free Seamless Knowledge browser from Ontopia. Powered by the Ontopia Knowledge Suite.

Welcome

Query results

Query:
occurrence (DNA_viral_genome, \$OCC)?

Execution time 0 millisecs
Result rows 3

OCC
[basic.Occurrence, 167 (URI http://psi.viral_genome.org/dna_viral_genome.wsd)]
[basic.Occurrence, 168 (URI http://psi.viral_genome.org/dna_viral_genome.asmx)]
[basic.Occurrence, 166 (URI uuid:73kv92bz-v0r5-gui2-sd89-bi28fuent841)]

Figure 5-9: tolog query results

5.6 XTM in comparison to alternative technologies

Although it is not always a sense in comparing things, in our case it may be of interest to look briefly at Topic Maps and OWL (as superseding RDF) in order to have a better idea how these interesting technologies 'compete' each other, but solving actually different problems.

First of all, Topic Maps and RDF have different origins: they are coming from two standard organizations - ISO, respectively W3C. This was maybe the point which determined some not enough informed parties to see them as competitors. Both technologies claim to approach easing the problem of infoglut, using knowledge representation techniques. In Topic Maps, the key-concept is the topic, created from the need of talking about *subjects*, while in RDF, the *resource* lies in the centre. It can then be said that the highlighted terms are playing the same role. Just that Topic Maps are knowledge representation from the perspective of humans, while RDF from the perspective of machines [Pep02].

The topics have in Topic Maps characteristics - names, occurrences and associations; in RDF, such things do not exist. In Topic Maps, the subjects out of whom topics are created, in order to make statements about them, can be addressable or non-addressable. The addressable subjects are identified by URIs and the non-addressable ones - by the URIs of the "subject indicators". The PSIs in Topic Maps facilitate interoperability across applications; in RDF such property does not exist. In RDF they are just resources, identified by URIs, but no other distinction. On the other side, the fact that PSIs are laying on URIs eases the interoperability for both technologies.

If we look at associations, the obvious aspect is that in Topic Maps we can have more than two "players", as RDF has. In other words, in Topic Maps, more than two roles can occur in an association. If we extend now the view to

OWL, regarding taxonomies, Topic Maps offer a simpler way to express same things that OWL could do, while OWL provides semantic validity. Also, some of the semantic foundations are identical in that Topic Maps do support class/subclass associations, although the class modelling in Topic Maps is not too strong. Topic Maps rely mostly on name-based classification (and merging), while OWL classes have robust properties, set theoretics and mapping. However, OWL and XTM are serializations with which one can ship information and have it arrived in a decipherable form at the other end, no matter which wire is used. Both bring to the table some kind of underlying process model and semantics. XTM makes a minimalist ontological commitment to object necessary to capture topics and subjects. To this, OWL adds some expressiveness, so it can be taken advantage of complementing the technologies [TMO04].

Finally, it can be said that XML Topic Maps found their place in the Semantic Web world and that there is no real competition between the mentioned technologies.

5.7 Summary

This extensive chapter introduced first the purpose of the case study and described briefly a few biological ontologies. It was shown that a topic map solution can act as a complete knowledge base with links to different information resources. This, of course, implies that our initial aim was reached, namely to base the discovery of Web Services on Semantic Web technologies.

Additionally, a simple methodology for developing XTM was defined, before starting the practical part. After the successful creation, three different ways in navigating and querying have been presented. The chapter concludes with a brief comparison of alternative technologies from the Semantic Web.

6 Results

The Semantic Web and Web Services are two new exciting research fields. In the centre of this work lays the concept of Topic Maps, more precisely of XML Topic Maps. These can, among others, be used for navigating enormous amounts of information. XTM do not support formal reasoning, but are excellent aids for helping people lost in information space.

In the world of Web Services, UDDI plays the role of the yellow pages, but unfortunately UDDI does not realize what it aims, because search capabilities within the registry are quite limited. The bourne of the thesis was to improve the discovery process of services by moving to more semantics. For this retrieval problem, we exploited the capabilities of XML Topic Maps. There exist a couple of efforts in combining Web Services with the Semantic Web in order to achieve a better search for Web Services. To my knowledge, currently such kind of implementation does not exist, with the aim to improve service discovery.

However, when talking about XTM, one talks simultaneously about organizing knowledge, more specifically about terms like taxonomy and ontology. These constructs consist of semantic terms and relations, whole communities agreed on. For this reason we used a part of the Gene Ontology as a basis for our prototype in the case study. The Topic Map was developed with the help of the Linear Topic Map notation and exported afterwards to XTM. The creation followed a “self made” methodology that turned out to be successful.

The result does not only improve semantic discovery with respect to Web Services. Additionally it was shown how XTM can be used for building a complete knowledge base. This enables a biologist to navigate through terms she/he already knows. To each term it was assigned a number of external information resources, like the unique identifier key of an UDDI entity, the

location of a WSDL file, a URL with additional information by a trustful organization (e.g. a bioinformatics institute), or even the location of figures relevant to the concept. That's why the final result of the thesis is doing much more than the title of the thesis promises.

It was also proven how to navigate and query a Topic Map with the help of the Omnigator software in several ways and tolog (a topic map query language) was briefly introduced.

Some possible problems should be considered; even the solution is quite satisfying. There is a continuous evolution of the Gene Ontology. This means that the XTM document has to be permanently maintained. Additionally the danger of dead links could occur, in case the XTM is not maintained properly (URNs could be solving this problem). If XTM should be used just for Web Services discovery in a domain (and not as a complete knowledge base), it is conceivable to refer directly to the location of the WSDL file, thus the stated problem to be solved completely without UDDI.

Note finally that the results out of this thesis can be used for all imaginable domains and not only for the bioinformatics domain, which was here just the realm of application.

References

- [AH04] Antoniou G., van Harmelen F., *A Semantic Web Primer*, London, MIT Press, 2004, ISBN 0-262-01210-3
- [AM03] Apte N., Mehta T., *UDDI – building registry-based web services solutions*, New Jersey, Prentice Hall Professional Technical Reference, 2003, pg. 395, ISBN 0-13-046457-0
- [And97] Anderson J., “Organization of Knowledge”, in: Feather J., Sturges P., *International Encyclopedia of Information and Library Science*, New York, Routledge, pg: 336, SCSU REF Z1006.157, 1997
- [Bie02] Biezunski M., “Introduction To The Topic Maps Paradigm”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [BMWS] Bada M., McEntire R., Wroe C., Stevens R., *GOAT: The Gene Ontology Annotation Tool*, www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/116.pdf
- [BN01] Biezunski M., Newcomb S. R., *XML Topic Maps: Finding Aids for the Web*, 2001, www.coolheads.com/SRNPUBS/ieee-mm-topicmaps-article.pdf
- [BSG04] Bada M., Stevens R., Goble C., et. al., “A short study on the success of the Gene Ontology”, *Journal of Web Semantics*, 2004, pg. 235-240

- [CN02] Carlstedt, A., Nordborg M., *An evaluation of Topic Maps*, A Master's Thesis in Computational Linguistics, Göteborg University, May 2002
- [DM05] Garshol L. M., Moore G., *Topic Maps – Data Model*, 2003, www.isotopicmaps.org/sam/sam-model/, edited: 2005-01-17, visited: 2005-05-12
- [Fen04] Fensel D., *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Berlin, Springer, 2004, ISBN 3-540-00302-9
- [Fre02] Freese E., “SemanText”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Gar05] Garshol L. M., private mail on 2005-01-19
- [GO04] An Introduction to Gene Ontology, www.geneontology.org/GO.doc.html, edited: 2005-01-11, visited: 2005-01-14
- [GOAT04] Gene Ontology Annotation Tool, goat.man.ac.uk/approach.html, edited: 2004-04-13, visited: 2005-01-14
- [GONG02] Gene Ontology Next Generation (GONG), gong.man.ac.uk/background.html, edited: 2003-03-07, visited: 2005-01-14
- [Gra02] Bénédicte le Grand, “Topic Map Visualization”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps*

-
- for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Gru93] Gruber, T. R., A translation approach to portable ontology specifications, *Knowledge Acquisition*, 5:199-220, 1993
- [Haw02] Hawke S., How the Semantic Web works, www.w3.org/2002/03/semweb/, edited: 2002-03-29, visited: 2004-12-12
- [Her04] Herman I., Questions (and Answers) on SW, www.w3.org/2004/Talks/0712-Brisbane-IH/11.html, Presentation on 2004-12-07, visited: 2005-01-06
- [Hun02] Hunting S., “How To Start Topic Mapping Right Away With The XTM Specification”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Liu04] Liu, Chih-Hsiung, Matchmaking based on OWL ontologies and Description Logics Inferences, www.im.ntu.edu.tw/IM/Theses/r92/R91725040.pdf, edited: 2004-07-12, visited: 2005-01-09
- [LTM02] Garshol L. M., The Linear Topic Map Notation, version 1.2, www.ontopia.net/download/ltn.html, edited: 2002-05-15, visited: 2002-01-20
- [Man03] Mann T. A.; *.NET Web Services for Dummies*; New York, Wiley Publishing, Inc., 2003, ISBN 0-7645-1647-7

- [Moo] Moore R., Data Management Services, swradio.omg.org/workshop/proceedings/Moore.ppt, no date of last edit, visited: 2005-01-11
- [NM00] Noy N., McGuinness D.; *Ontology Development 101: A Guide to Creating Your First Ontology*; Stanford University, 2000, protege.stanford.edu/publications/ontology_development/ontology_101-noy-mcguinness.html, edited: 2005-05-13, visited: 2005-05-13
- [Oas03] *OASIS - Published Subjects: Introduction and Basic Requirements*, www.ontopia.net/tmp/pubsubj-gentle-intro.htm, edited: 2003-06-24, visited: 2005-02-07
- [Omn05] omnigator eight, www.ontopia.net/omnigator/models/index.jsp, edited: 2005-02-19, visited: 2005-02-19
- [Ont] *Welcome to the Land of Ontopia*, www.ontopia.net/, no date of last edit, visited: 2004-12-13
- [Ont04] The Ontopia Omnigator “User’s Guide”, www.ontopia.net/omnigator/userguide.html, edited: 2004-12-09, visited: 2005-01-15
- [OL02] Obrst L., Liu H., “Knowledge Representation, Ontological Engineering, And Topic Maps”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Oli00] Oliveira A., *Deployment of Topic Maps for Navigation and Searching in Huge Information Spaces as Component of Learning Environment*, 2000, www.ccg.pt/Publications/_PDFs/Theses/2000/Diploma_Thesis.pdf, edited: 2004-11-12, visited: 2004-11-12

- [OWL05] Introduction to OWL, www.w3schools.com/rdf/rdf_owl.asp,
edited: 2005-01-12, visited: 2005-01-12
- [Par02] Park J., "Let There Be Light", in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Par04] Park J., private mail on 2004-08-12
- [Pep02] Pepper S., *Ten Theses on Topic Maps and RDF*,
www.ontopia.net/topicmaps/materials/rdf.html,
edited: 2003-04-15, visited: 2004-12-14
- [Pep04] Pepper S., *The TAO of Topic Maps - Finding the way in the age of infoglut*, www.ontopia.net/topicmaps/materials/tao.html,
edited: 2004-04-20, visited: 2004-07-07
- [PG04] Pepper S., Garshol L. M., *The XML Papers - Lessons on Applying Topic Maps*,
www.ontopia.net/topicmaps/materials/xmlconf.html,
edited: 2004-04-20, visited: 2004-12-13
- [Rij96] Rijsbergen C., Information Retrieval,
www.dcs.gla.ac.uk/Keith/Chapter.1/Ch.1.html,
edited: 1996-09-18, visited: 2004-12-12
- [RM03] Newcomb S. R., Hunting S., Algermissen J., Durusau P.,
Topic Maps – Reference Model, ISO/IEC JTC 1/SC34,
www.isotopicmaps.org/tmrm/TMRM-3.10/TMRM-3.10.html,
December 2003, edited: 2003-12-02, visited: 2004-08-26

- [RN03] Russel S., Norvig P. *Artificial Intelligence, A modern approach*, New Jersey, Alan R. Apt, 2003, ISBN 0-13-080302-2
- [SGB00] Stevens R., Goble A. C., Bechhofer S., *Ontology-based Knowledge Representation for Bioinformatics*, University of Manchester,
www.cs.man.ac.uk/~stevensr/papers/briefings-ontology.pdf,
edited: 2000-09-26, visited: 2004-12-16
- [Sig02] Sigel A., “Topic Maps in Knowledge Organization”, in: Park J., Hunting S., *XML Topic Maps, Creating and using Topic Maps for the Web*, Boston, Addison Wesley, 2002, ISBN 0-201-74960-2
- [Sow00] Sowa J., *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks-Cole, Pacific Grove, 2000, ISBN 0-534-94965-7, www.jfsowa.com/krbook/,
edited: 2004-01-12, visited: 2004-11-11
- [SVSM03] Sivashanamugan K., Verma K., Sheth A., Miller J., *Adding Semantics to Web Services Standards*, University of Georgia, lstdis.cs.uga.edu/library/download/SVSM03-ICWS-final.pdf,
no last edit, visited: 2004-11-13
- [Swa02] Swartz A., The Semantic Web for Developers,
logicerror.com/semanticWeb-webdev, edited: 2002-05-01,
visited: 2004-12-12
- [TA98] TAMBIS, imgproj.cs.man.ac.uk/tambis/index.html,
edited: 2004-03-14, visited: 2005-01-14

-
- [Tax03] searchsmallbizit.techtarget.com/sDefinition/0,,sid44_gci331416,00.html, edited: 2004-11-12, visited: 2004-11-12
- [TMO04] ontolog forum, “Topic Maps and OWL”, ontolog.cim3.net/forum/ontolog-forum/2004-01/msg00014.html, edited: 2004-01-02, visited: 2005-01-17
- [Tol04] tolog – Language Tutorial, www.ontopia.net/omnigator/docs/query/tutorial.html, edited: 2004-12-08, visited: 2005-03-11
- [Vos05] Voskob M., Taxonomies and Value Sets, www.oasis-open.org/committees/download.php/5322/uddi-spec-tc-taxonomies-discussion-20040205.doc, edited: January 2005, visited: 2005-01-13
- [XTM01] XML Topic Maps (XTM) 1.0 – TopicMaps.Org Specification, www.topicmaps.org/xtm/1.0/, edited: 2005-01-13, visited: 2005-01-28
- [WS02] Web Services - An Introduction, www.webservicesmall.com/profile.asp?pid=65, no date of last edit, visited: 2004-11-12
- [WS04] Web Services Architecture W3C Working Group Note, Web Services Definition, www.w3.org/TR/2004/NOTE-ws-arch-20040211/, edited: 2004-02-11, visited: 2004-10-29
- [Zar04] Zarvic N., *Master Thesis: Consuming existing Web Services for bioinformatics*, University of Osnabrück and University of Twente, 2004

- [ZTP03] Zimmermann O., Tomlinson M., Peuser S., *Perspectives on Web Services – Applying SOAP, WSDL and UDDI to Real-World Projects*, Berlin, Springer, 2003, ISBN 3-540-00914-0

Glossary

API

Application Programming Interface is the structure allowing to program to a pre-constructed interface instead of programming directly a device or a piece of software.

CERN

The European Organization for Nuclear Research where the web was born. The original name in French was Conseil Europeen pour la Recherche Nucleaire.

DAML

DARPA Agent Markup Language is a markup language for the U.S. Defense Advanced Research Project Agency that is based on XML; DAML is designed to have a better capacity than XML for describing objects and the relationships between them, to express semantics, to create a higher level of interoperability among Web sites.

DNA

deoxyribonucleid acid, usually 2'-deoxy-5'-ribonucleic acid; it is a code used within cells to form proteins.

HyTime

Hypermedia/Time-based Structuring Language (ISO/IEC 10744:1992) is the international standard for the platform-neutral representation of hypertext and multimedia information; an extension of SGML.

ISO

International Organization for Standardization; ISO is not an acronym; the name derives from the greek word *isō*, which means equal. Funded in 1946, ISO is composed of national standards bodies from over 75 countries.

KB

Knowledge Base is a collection of procedures, facts, rules, organized into schemas; used in this work with the sense of kind of registry which refers to all resources relevant to the topic.

LTM

Linear Topic Map notation is a simple textual format for Topic Maps.

OASIS

Organization for the Advancement of Structured Information System (formerly known as SGML Open) is an international consortium whose aim is to promote the adoption of product-independent standards for information formats such as SGML, XML and HTML.

OWL

Web Ontology Language is a markup language for publishing and sharing data using ontologies on the Internet; it is derived from the DAML+OIL and is a vocabulary extension of RDF. Together with RDF and other components, these tools make up the semantic web project.

PSI

A subject indicator that is maintained at an advertised address to serve as a subject identity point for topics in topic maps created by anyone; Published Subject Indicators should unambiguously and compellingly indicate their subjects.

RDF

The Resource Description Framework is a general framework for how to describe any Internet resource such as Web site and its content. An RDF description (often referred to as *metadata*, or 'data about data') can include the authors of the resource, date of creation or updating, keywords for searching engine data collection, etc.

RDFS

Resource Description Framework Schema is a universal language that lets users describe resources using their own vocabularies. RDF does not make assumptions about any particular application domain and does not define the semantics of any domain. It is up to the user to do so in RDFS.

SGML

Standard Generalized Markup Language; it was adopted in 1986 as an international standard (ISO 8879) for the creation, storage, delivery and management of information products. HTML and its possible successor, XML, are both subsets of SGML.

SOAP

It is an XML-based protocol for exchange of information in a decentralized, distributed environment.

TAMBIS

Transparent Access to Multiple Biological Information Sources; TAMBIS project aims to provide transparent access to disparate biological databases and analysis tools; enables users to utilize a wide range of resources with the minimum of effort.

TMRM

The Topic Map - Reference Model provides a systematic way to make the subjects and relationships explicit. TMRM does not extend ISO 13250:2002, but provides an information model that enables meaningful construction, description, comparison and evaluation of syntaxes, data models.

UDDI

Universal Discovery Description and Integration is a platform-independent, XML-based registry for businesses all over the world to be listed on the Internet; it is an open industry initiative (sponsored by OASIS) enabling businesses to discover each other and define how to interact over the Internet.

UML

Unified Modelling Language is a notational language for specifying and visualizing complex software, especially large, object-oriented projects.

URI

Uniform Resource Identifier – the addressing technology which identifies resources in Internet or a private intranet; it originally defined as two types: URL (address with network locations) and URN (Uniform Resource Name), (persistent name, address independent).

URL

Uniform Resource Locator – an Internet address (e.g., *http://www.ontopia.net/topicmaps/*), usually consisting of the access protocol (*http*), the domain name (*www.ontopia.net*), and optionally the path to a resource or a file residing on that server (*topicmaps*).

UUID

The Universal Unique Identifier is a 128-bit number used to identify uniquely some entity or *object* in Internet; it relies upon a combination of components to ensure uniqueness.

WSDL

The Web Services Description Language is an XML format for describing network services as a set of endpoints operating on messages which contain document-oriented or procedure-oriented information.

XLink

XML Linking Language allows elements to be inserted into XML documents for creating and describing links between resources. It uses XML syntax for creating structures that can describe links similarly to the unidirectional hyperlinks of HTML, but other more sophisticated links.

XML

eXtensible Markup Language – metalanguage written in SGML that allows using to allow for easy interchange of documents on the World Wide Web.

W3C

World Wide Web Consortium was created in 1994 for leading the World Wide Web to its full potential by developing common protocols which promote its evolution and ensure its interoperability.

Appendix A

A simple biological classification in XTM

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<topicMap id="biotm"
  xmlns="http://www.topicmaps.org/xtm/1.0/"
  xmlns:xlink="http://www.w3.org/1999/xlink">

<!-- ..... ONTOLOGY TOPICS ..... -->
<!-- ..... THE TOPIC MAP TOPIC ..... -->

<topic id="biotm-topic">
  <!-- Reifies the topic map and gives it a name -->
  <subjectIdentity>
    <subjectIndicatorRef xlink:href="#biotm"/>
  </subjectIdentity>
  <baseName>
    <baseNameString>Biomolecule Topic
Map</baseNameString>
  </baseName>
  <baseName>
    <scope>
      <subjectIndicatorRef
xlink:href="http://psi.ontopia.net/basename/#short-
name"/>
    </scope>
    <baseNameString>Biomolecule TM</baseNameString>
  </baseName>
</topic>

<topic id="Biomolecule">
  <baseName>
    <baseNameString>Biomolecule</baseNameString>
  </baseName>
</topic>

<topic id="Protein">
  <baseName>
    <baseNameString>Protein</baseNameString>
  </baseName>
</topic>

<topic id="DNA">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
"http://www.rothamsted.bbsrc.ac.uk/notebook/courses/
guide/images/dna.gif"/>
  </subjectIdentity>
```

```
<baseName>
  <scope>
    <topicRef xlink:href="#EN"/>
  </scope>
  <baseNameString>DNA</baseNameString>
</baseName>
<baseName>
  <scope>
    <topicRef xlink:href="#DE"/>
  </scope>
  <baseNameString>DNS</baseNameString>
</baseName>
<occurrence>
  <resourceRef xlink:href="DNA.gif"/>
</occurrence>
</topic>

<topic id="EN">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
      "http://www.topicmaps.org/xtm/1.0/language.xtm#en"/>
  </subjectIdentity>
  <baseName>
    <baseNameString>En</baseNameString>
  </baseName>
</topic>

<topic id="DE">
  <subjectIdentity>
    <subjectIndicatorRef xlink:href=
      "http://www.topicmaps.org/xtm/1.0/language.xtm#de"/>
  </subjectIdentity>
  <baseName>
    <baseNameString>De</baseNameString>
  </baseName>
</topic>

<!-- ..... ASSOCIATIONS ..... -->

<association id="DNA-Biomolecule-association">
  <instanceOf>
    <subjectIndicatorRef xlink:href=
      "http://www.topicmaps.org/xtm/1.0/core.xtm#class-
instance"/>
  </instanceOf>
  <member>
    <roleSpec>
      <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#instance"/>
    </roleSpec>
    <topicRef xlink:href="#DNA"/>
  </member>
  <member>
    <roleSpec>
```

```
        <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class"/>
        </roleSpec>
        <topicRef xlink:href="#Biomolecule"/>
    </member>
</association>

<association id="Protein-Biomolecule-association">
    <instanceOf>
        <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class-
instance"/>
    </instanceOf>
    <member>
        <roleSpec>
            <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#instance"/>
            </roleSpec>
            <topicRef xlink:href="#Protein"/>
        </member>
        <member>
            <roleSpec>
                <subjectIndicatorRef xlink:href=
"http://www.topicmaps.org/xtm/1.0/core.xtm#class"/>
                </roleSpec>
                <topicRef xlink:href="#Biomolecule"/>
            </member>
        </association>

</topicMap>
```

Appendix B

LTM document developed manually in section 5.4

```

#TOPICMAP cell-tm
#INCLUDE "dc.ltm"

/* ONTOLOGY TOPICS */

/* THE TOPIC MAP TOPIC */
[celltm-topic = "Topic Map of a Cellular Component from
the Gene Ontology" /short @"#cell-tm"]

/* THEMES (SCOPING TOPICS) */
[short          = "short name"
@"http://psi.ontopia.net/basename/#short-name"]

/* TOPIC TYPES */
[ontology       = "Ontology"; "structured controlled
vocabulary"]
[viral          = "Viral"]
[cellular       = "Cellular"]

/* OCCURRENCE TYPES */
[description    = "Description"

@"http://psi.ontopia.net/xtm/occurrence-
type/description"]
[registry       = "Registry"
@"http://psi.ontopia.net/xtm/occurrence-type/registry"]
[service        = "Service"
@"http://psi.ontopia.net/xtm/occurrence-type/service"]
[figure         = "Figure"
@"http://psi.ontopia.net/xtm/occurrence-type/figure"]
[website        = "Website"]

/* ASSOCIATION TYPES */
[is-a           = "Is a"]
[part-of        = "Part of"]

/* TOPICS, ASSOCIATIONS AND OCCURRENCES */
[cellular_component = "Cellular Component"]

[virion : cellular = "Virion"]
{virion, figure,
"http://www.illumann.de/illumann%20Site/Handgemachtes
/Virion.jpg"}

[viral_capsid : viral = "Viral capsid"]

```

```

{viral_capsid, description,
"http://psi.virion.org/viral_capsid.wsdl"}
{viral_capsid, registry,
"uuid:8f156b70-bca8-4fac-90cd-820c42dc2e39"}
{viral_capsid, service,
"http://psi.virion.org/viral_capsid.asmx"}

[capsomere          = "Capsomere"]
{capsomere, description,
"http://psi.viral_capsid.org/capsomere.wsdl"}
{capsomere, registry,
"uuid:8f156m10-car9-2sam-83ud-159b59vu2e48"}
{capsomere, service,
"http://psi.viral_capsid.org/capsomere.asmx"}

[helical_viral_capsid      = "Helical viral capsid"]
{helical_viral_capsid, description,
"http://psi.viral_capsid.org/helical_viral_capsid.wsdl"}
{helical_viral_capsid, registry,
"uuid:1d754s34-smr7-8mev-49ks-907w25mw3r27"}
{helical_viral_capsid, service,
"http://psi.viral_capsid.org/helical_viral_capsid.asmx"}

[icosahedral_viral_capsid = "Icosahedral viral capsid"]
{icosahedral_viral_capsid, description,
"http://psi.viral_capsid.org/
icosahedral_viral_capsid.wsdl"}
{icosahedral_viral_capsid, registry,
"uuid:3h374d08-kdi8-6mel-91ne-649h65as9e37"}
{icosahedral_viral_capsid, service,
"http://psi.viral_capsid.org/
icosahedral_viral_capsid.pl"}

[viral_capsid_sensu_Retroviridae      = "Viral capsid
sensu Retroviridae"; / sensu Retroviridae]
{viral_capsid_sensu_Retroviridae, description,
"http://psi.viral_capsid.org/
viral_capsid_sensu_retroviridae.wsdl"}
{viral_capsid_sensu_Retroviridae, registry,
"uuid:hr49nyi2-5rf8-m2u1-aje2-nz5so30ba71i"}
{viral_capsid_sensu_Retroviridae, service,
"http://psi.viral_capsid.org/viral_capsid_sensu_retroviri
dae.pl"}

[viral_nucleocapsid          = "Viral nucleocapsid"]
{viral_nucleocapsid, description,
"http://psi.viral_capsid_sensu_retroviridae/
viral_nucleocapsid.wsdl"}
{viral_nucleocapsid, registry,
"uuid:8m57uxmw-m7z0-d21t-m32r-mwo7l35sxi5j"}
{viral_nucleocapsid, service,
"http://psi.viral_capsid_sensu_retroviridae/
viral_nucleocapsid.pl"}

```



```
[viral_portal_complex      = "Viral portal complex"]
{viral_portal_complex, description,
"http://psi.viral_capsid.org/viral_portal_complex.wsdl"}
{viral_portal_complex, registry,
"uuid:nri52d9b-286h-13f8-ew97-me62vakr1571"}
{viral_portal_complex, service,
"http://psi.viral_capsid.org/viral_portal_complex.asmx"}

[viral_scaffold           = "Viral scaffold"]
{viral_scaffold, description,
"http://psi.viral_capsid.org/viral_scaffold.wsdl"}
{viral_scaffold, registry,
"uuid:neiv472h-cas6-f24q-ny12-nw3079ks4dvz"}
{viral_scaffold, service,
"http://psi.viral_capsid.org/viral_scaffold.asmx"}

[viral_envelope           = "Viral envelope"]
{viral_envelope, description,
"http://psi.virion.org/viral_envelope.wsdl"}
{viral_envelope, registry,
"uuid:bw70k81s-nw5b-qp4q-5w2l-ndox815hs2u6"}
{viral_envelope, service,
"http://psi.virion.org/viral_envelope.pl"}

[viral_genome              = "Viral genome"]
{viral_genome, description,
"http://psi.virion.org/viral_genome.wsdl"}
{viral_genome, registry,
"uuid:w94nfo18-vjr6-v8s0-fibe-bsu62k5n39bp"}
{viral_genome, service,
"http://psi.virion.org/viral_genome.pl"}

[provirus : viral         = "Provirus"]
{provirus, description,
"http://psi.viral_genome.org/provirus.wsdl"}
{provirus, registry,
"uuid:bd7vo3ms-93m6-20gr-9d21-vi3850f86kba"}
{provirus, service,
"http://psi.viral_genome.org/provirus.asmx"}

[DNA_viral_genome         = "DNA viral genome"]
{DNA_viral_genome, description,
"http://psi.viral_genome.org/dna_viral_genome.wsdl"}
{DNA_viral_genome, registry,
"uuid:73kv92bz-v8r5-gui2-sd89-bi28fuent841"}
{DNA_viral_genome, service,
"http://psi.viral_genome.org/dna_viral_genome.asmx"}

[non-segmented_viral_genome = "Non-segmented viral
genome"]
{non-segmented_viral_genome, description,
"http://psi.viral_genome.org/
nonsegmented_viral_genome.wsdl"}
{non-segmented_viral_genome, registry,
```

```
"uuid:zxr65ld8-59bw-br29-7x4m-hsprm726vt7d"}
{non-segmented_viral_genome, service,
"http://psi.viral_genome.org/
nonsegmented_viral_genome.pl"}

[RNA_viral_genome          = "RNA viral genome"]
{RNA_viral_genome, description,
"http://psi.viral_genome.org/rna_viral_genome.wsdl"}
{RNA_viral_genome, registry,
"uuid:f8nr5xir-4n6g-v72d-f92n-0hsut735mt7g"}
{RNA_viral_genome, service,
"http://psi.viral_genome.org/rna_viral_genome.pl"}

[dsRNA_viral_genome       = "dsRNA viral genome"]
{dsRNA_viral_genome, description,
"http://psi.rna_viral_genome.org/
dsrna_viral_genome.wsdl"}
{dsRNA_viral_genome, registry,
"uuid:jr8vz56v-g8e7-578n-sh5g-nf75nsh58gn3"}
{dsRNA_viral_genome, service,
"http://psi.rna_viral_genome.org/dsrna_viral_genome.pl"}

[ssRNA_viral_genome       = "ssRNA viral genome"]
{ssRNA_viral_genome, description,
"http://psi.rna_viral_genome.org/
ssrna_viral_genome.wsdl"}
{ssRNA_viral_genome, registry,
"uuid:ndz6va54-vz4n-792b-cg50-mst50bh28hc3"}
{ssRNA_viral_genome, service,
"http://psi.rna_viral_genome.org/
ssrna_viral_genome.asmx"}

[ambisense_viral_genome   = "Ambisense viral genome"]
{ambisense_viral_genome, description,
"http://psi.ssrna_viral_genome.org/
ambisense_viral_genome.wsdl"}
{ambisense_viral_genome, registry,
"uuid:dut63n59-v6fk-x53l-cg69-nd7gw3lv84ng"}
{ambisense_viral_genome, service,
"http://psi.ssrna_viral_genome.org/
ambisense_viral_genome.asmx"}

[negative_sense_viral_genome = "Negative sense viral
genome"]
{negative_sense_viral_genome, description,
"http://psi.ssrna_viral_genome.org/
negative_sense_viral_genome.wsdl"}
{negative_sense_viral_genome, registry,
"uuid:vzsnf759-s5h7-v36m-19f4-ns64hviw7x61"}
{negative_sense_viral_genome, service,
"http://psi.ssrna_viral_genome.org/
negative_sense_viral_genome.pl"}
```

```
[positive_sense_viral_genome      = "Positive sense viral
genome"]
{positive_sense_viral_genome, description,
"http://psi.ssrna_viral_genome.org/
positive_sense_viral_genome.wsdl"}
{positive_sense_viral_genome, registry,
"uuid:vum48v62-vg68-4h8j-t7b2-fr7me68m4g5f"}
{positive_sense_viral_genome, service,
"http://psi.ssrna_viral_genome.org/
positive_sense_viral_genome.asmx"}

[segmented_viral_genome      = "Segmented viral genome"]
{segmented_viral_genome, description,
"http://psi.viral_genome.org/
segmented_viral_genome.wsdl"}
{segmented_viral_genome, registry,
"uuid:d6m72gmk-e6b7-v68b-h1g0-n30xzemni267"}
{segmented_viral_genome, service,
"http://psi.viral_genome.org/segmented_viral_genome.pl"}

[bipartite_viral_genome      = "Bipartite viral genome"]
{bipartite_viral_genome, description,
"http://psi.segmented_viral_genome.org/
bipartite_viral_genome.wsdl"}
{bipartite_viral_genome, registry,
"uuid:vue75m39-e5h6-2g65-sa12-19785rff1000"}
{bipartite_viral_genome, service,
"http://psi.segmented_viral_genome.org/
bipartite_viral_genome.pl"}

[multipartite_viral_genome = "Multipartite viral genome"]
{multipartite_viral_genome, description,
"http://psi.segmented_viral_genome.org/
multipartite_viral_genome.wsdl"}
{multipartite_viral_genome, registry,
"uuid:d5h5t5ii-60y2-5gas-gq9t-7bqu6d3901g3"}
{multipartite_viral_genome, service,
"http://psi.segmented_viral_genome.org/
multipartite_viral_genome.pl"}

[tripartite_viral_genome      = "Tripartite viral genome"]
{tripartite_viral_genome, description,
"http://psi.segmented_viral_genome.org/
tripartite_viral_genome.wsdl"}
{tripartite_viral_genome, registry,
"uuid:d5jwu245-5mni-51cw-90nq-vuw75mxuw067"}
{tripartite_viral_genome, service,
"http://psi.segmented_viral_genome.org/
tripartite_viral_genome.asmx"}

[viral_procapsid : viral      = "Viral procapsid"]
{viral_procapsid, description,
"http://psi.virion.org/viral_procapsid.wsdl"}
{viral_procapsid, registry,
```

```

"uuid:39viwnz6-103b-baio-vz30-vwmt05n7x730"}
{viral_procapsid, service,
"http://psi.virion.org/viral_procapsid.asmx"}

[viral_tegument      = "Viral tegument"]
{viral_tegument, description,
"http://psi.virion.org/viral_tegument.wsd1"}
{viral_tegument, registry,
"uuid:v73n80x3-4n81-cc91-7bao-vi2ns74n6x02"}
{viral_tegument, service,
"http://psi.virion.org/viral_tegument.pl"}

[gene_ontology : ontology = "The Gene Ontology";
                        "Gene Ontology, the"
@"http://www.geneontology.org/ontology/
gene_ontology.obo"]
{gene_ontology, ontology,
"http://www.geneontology.org/ontology/gene_ontology.obo"}

is-a(virion, cellular_component)
is-a(helical_viral_capsid, viral_capsid)
is-a(icosahedral_viral_capsid, viral_capsid)
is-a(viral_capsid_sensu_Retroviridae, viral_capsid)
is-a(provirus, viral_genome)
is-a(DNA_viral_genome, viral_genome)
is-a(non-segmented_viral_genome, viral_genome)
is-a(RNA_viral_genome, viral_genome)
is-a(dsRNA_viral_genome, RNA_viral_genome)
is-a(ssRNA_viral_genome, RNA_viral_genome)
is-a(ambisense_viral_genome, ssRNA_viral_genome)
is-a(negative_sense_viral_genome, ssRNA_viral_genome)
is-a(positive_sense_viral_genome, ssRNA_viral_genome)
is-a(segmented_viral_genome, viral_genome)
is-a(bipartite_viral_genome, segmented_viral_genome)
is-a(multipartite_viral_genome, segmented_viral_genome)
is-a(tripartite_viral_genome, segmented_viral_genome)

part-of(viral_capsid, virion)
part-of(capsomere, viral_capsid)
part-of(viral_nucleocapsid,
viral_capsid_sensu_Retroviridae)
part-of(viral_portal_complex, viral_capsid)
part-of(viral_scaffold, viral_capsid)
part-of(viral_envelope, virion)
part-of(viral_genome, virion)
part-of(viral_procapsid, virion)
part-of(viral_tegument, virion)

/* ADDRESSABLE SUBJECT */
[gene_ontology-website : website = "Gene Ontology's
Website" %"http://www.geneontology.org/"]

```

Herewith I affirm that I have written the thesis independently. I have used only the sources and aids mentioned in the thesis.

Osnabrück, _____

Signature