An RDF Schema for Thesauri

Student: Markus Klie (2211522)

Supervisors: Roel Teeninga, Peter Wuebbelt

Deventer, June 6th, 2003

Information Services and Management 2002/2003

Hereby I testify that everything in this dissertation is written by my own - except the printed or electronic sources I have declared as a citation or as a paraphrase and identified by all the needed bibliographic information. I know that my dissertation cannot be accepted with undeclared sources and that a plagiarism will cause that I fail the graduation.

Deventer, June 6th, 2003

(Markus Klie)

I would like to thank O'Reilly. Your books were my study.

Table of Contents

Abstract

In this dissertation we demonstrate that thesauri, as definded by information science, can be represented by means of the vocabulary description language of the Resource Description Framework (RDF). We explain the theory of thesauri, point out the shortcomings of native XML representations and develop an RDF schema that we compare with the result of an similar attempt of the CERES/NBII project. An RDF Schema for Thesauri serves the integration of independently evolved thesauri into one controlled vocabulary. Institutions that have to integrate data that were described with different thesauri or institutions that want to make use of more than one thesaurus in order to describe their documents can benefit from our schema.

PART 1

Introduction

Imagine a librarian sitting at a desk that is occupied by a card-index box and two stacks of books - one on his left, one on his right. He takes the uppermost book of the left stack, reads the title and scans table of contents and foreword. He checks a list of words for a term that describes the content of the book in a proper way and writes down title and shelfmark of this book on the index card of this term. Afterwards he puts the book atop the stack on his right and continues with the next book on his left. Thanks to his work a user who is searching for a book without knowing title and author can find it by checking the index card of the appropriate term.

Imagine the Web with its literally billions of websites: There is neither a librarian who is responsible for describing the content of the websites nor a word list that might serve this purpose; and it would be naive if we believed that one day there might be both. The content and the purpose of the websites are too heterogeneous, the interests of their creators and users too different in order to hope that one day we will succeed in developing a word list that is accepted worldwide.

Instead of trying to develop an online counterpart of our librarian's word list, wise people decided to develop a standard that allows us to develop our own vocabulary with that we can describe our websites. The crucial point is that all vocabularies that are described by this standard can be connected with each other. In this way we can build a word list by combining already existing ones instead of having to start from the scratch.

Our examples seem to originate from two totally different worlds: the world of our librarian seems to be small and clear in contrast to the world of the Web. But is it? Is there on principle a difference between describing the content of a book and the content of a website? And what about the other librarians working in literally thousands of other libraries that make use of different word lists? Couldn't we use the standard that was developed in order to solve one of the latest problems in order to solve one that has existed for a long time but that we are just now becoming aware of?

In the course of this dissertation we are going to answer this question by first analyzing the theory behind the word lists (also known as thesauri) and second the theory behind the standard for developing vocabularies for describing online resources (als known as RDF). By doing so, we want to find out whether it is possible to describe the terms and their relationships of a thesauri with the help of RDF and why RDF is superior to other possibilities like XML. Our aim is to develop a standard for describing thesauri by means of RDF.

By means of such a standard libraries could first describe their thesauri and second integrate their thesauri into one metathesaurus that could enable users to search several databases of libraries with a single keyword although the particular catalogues make use of different keywords for the same purpose. The application of such a networked vocabulary is of course not confined to the description of books in libraries but rather it can be also used to describe all kinds of documents among them for instance newspaper articles, software and - as a matter of course - websites.

PART 2

Thesauri

2.1 What is a Thesaurus?

Suppose you have a database that stores articles about one of your hobbies - say cars. In order to facilitate the search for articles you start to assign each article a subject heading like repair, sportscar, tires, cabrio, fuel consumption, BMW, varnishing. The more articles your database contains the more subject headings you will create. One day you might feel the need for organising your subject headings by grouping them by brand names, models, parts and processes. A possible subject heading belonging to the group of parts could be engine for articles about engines in general; another one could be Diesel engine for articles about Diesel engines in particular. You feel somehow that Diesel engine is subordinated to engine. Supposing that your friends have an increasing interest in using your database, you might face the situation that you and your friends use different terms for the same thing. Without being aware of it, you are about to construct a thesaurus since a thesaurus is nothing but a hierarchically arranged list of subject headings.

The main purpose of a thesauri is to control the language of a defined user group in order to index and retrieve relevant documents [1] for this user group in a consistent way. A thesaurus is only one among several tools that serve this purpose. In general we call a deliberately arranged language that is meant for indexing documents an indexing language. There are mainly two kinds of indexing languages: classification schemes and controlled vocabularies. A detailed discussion of classification schemes is beyond the scope of this dissertation [2]. But it is important to note that a classification scheme is a very strict hierarchy of categories in that subcategories are required by definition to reflect all aspects of the superordinated category without overlapping each other. In contrast to that, controlled vocabularies like thesauri and subject heading lists merely group terms according to their sematics. Although thesauri do this also in a hierarchical way, the arrangement is by far not as strict as a classification.

Since thesauri are used for a long time the theory behind them is well-known and written down in various standards, among them ISO 2788:1986 [3], the standard of the International Organisation for Standardisation (ISO); Z39.19 [4], the standard of the National Information Standards Organisation (NISO), approved by the American National Standards Institute

[1] Note that in information science every unit that has to be described is considered a document, e.g. a book, a magazine, an article of a magazine or a website. It is up to the indexing authority to decide whether and to what extent a particular item is decomposed into documents, i.e. if only the magazine is described or every article of this magazine.

[2] However, classifications and RDF should be subject of another dissertation.

[3] cf. ISO 2788, 1986

[4] cf. ANSI/NISO Z.39.19, 1993

(ANSI) and DIN 1463-1 [5], the standard of the Deutsches Institut für Normung (DIN). Thanks to the efforts of the ISO there are essentially no differences among these standards when it comes to the basic theory. In the course of this dissertation we will mostly refer to Z39.19 since it excels in exemplary descriptions and definitions, among them the formal definition of a thesaurus:

> "[A thesaurus is] a controlled vocabulary arranged in a known order in which equivalence, homographic, hierarchical, and associative relationships among terms are clearly displayed and identified by standardized relationship indicators, which must be employed reciprocally. Its purposes are to promote consistency in the indexing of documents, predominantly for postcoordinated information storage and retrieval systems, and to facilitate searching by linking entry terms with descriptors. Thesauri may also facilitate the retrieval of documents in free text searching". [6]

2.2 Terms and Concepts

One of the first steps of thesaurus construction is the compilation of terms used by the users. After having successfully scanned hundred of pages the developers of a thesaurus face the situation that they have to arrange what they have found. For simplicity let us assume that the result of the first step is a list consisting merely of the following terms [7]:

Accounting
Adolescents
Bookkeeping
Bribery
Corruption
Courses of study
Curriculum
Syllabus
Teenagers
Youth

After having discussed for some time the involved person agree that some of these terms mean the same thing. The terms are grouped accordingly:

Accounting
Bookkeeping

Corruption
Bribery

Curriculum
Courses of study
Syllabus

[5] cf. DIN 1463-1, 1987

[6] ANSI/NISO Z.39.19, 1993, p. 38

[7] cf. thesaurus of the Australian Public Affairs Information Service; APAIS, 2002

Youth
Adolescents
Teenagers

Instead of saying that for instance Accounting and Bookkeeping in fact mean the same *thing* - for this sounds a bit too colloquial, especially in the context of a dissertation - professionals say that Accounting and Bookkeeping belong to the same *concept*. Z39.19 defines a concept as follows:

> "A unit of thought, formed by mentally combining some or all of the characteristics of a concrete or abstract, real or imaginery object. Concepts exist in the mind as abstract entities independent of terms used to express them." [8]

However this definition leads us into temptation to discuss the concept of a concept from a philosophical point of view we should confine ourselves to understand the basic meaning: a concept stands for a concrete or abstract thing that can be usually described by some terms that are more or less equivalent. The concept itself is independent of its terms. In the majority of thesauri one of the terms of a concept is chosen as the preferred term (also known as descriptor) while the remaining terms function as entry terms referring to this preferred term. These references are usually expressed reciprocally by means of the relationship indicators use (USE) and used for (UF):

Youth
UF Adolescents
UF Teenagers

Adolescents
USE Youth

Teenagers
USE Youth

Only preferred terms are used to describe documents, the purpose of the entry terms is to lead the user to the appropriate descriptor. That is why entry terms are sometimes referred to as lead-in terms.

As as concept is independent of its terms we should discuss whether it is recommended to explicitly represent a concept. Let us consider an example of a concept and its terms [9]:

Disadvantaged groups
UF Deprived groups
UF Underprivileged

All three terms are belonging to the same concept. Obviously the developers of this thesaurus have decided that Disadvantaged groups is the preferred term that is referred to by Deprived

[8] ANSI/NISO Z.39.19, 1993, p. 35

[9] cf. APAIS, 2002

groups and Underprivileged. Since it would have been also acceptable if they had chosen Underprivileged as preferred term this can be considered a rather arbitrary decision.

As mentioned above, hierarchical relationships exist only among preferred terms. As a matter of course Disadvantaged groups is also part of a hierarchy:

Disadvanaged groups
BT Social problems

Accordingly we have to mention Disadvantaged groups as narrower term of Social problems:

Social problems
NT Aged abuse
NT Alcohol abuse
NT Child abuse
NT Crime
NT Disadvantaged groups
NT Discrimination
NT Divorce
NT Drug abuse
NT Poverty
NT Suicide
NT Violence

Although human nature will probably cater for the persistence of this concept until doomsday it is very likely that the preferred term for this concept will change in the course of time. Maybe Disadvanged groups will be considered politically incorrect in the future and therefore has to become an entry term - possibly referring to a newly created term called Socially challenged. Although the concept remains the same the preferred term would change and thus all relationships within the thesaurus referring to the old preferred term would have to be updated. In case of the APAIS thesaurus Disadvantaged groups is in total referred to 6 times - including the associative relationship that we will discuss below. That is why some critics argue that hierarchical relationships should be only established among concepts, but not among terms [10]. But in order to do so we have to explicitly represent a concept. Usually this is done by a meaningless identifier like a unique number:

Concept 4711
Disadvantaged groups (Preferred)
Deprived groups
Underpriviledged

In spite of being aware of the distinction between a concept and its terms, it is usually hard to discuss a concept without mentioning a term. Nobody would say for example "We have to link concept 4711 with concept 0815". Without compromising our precision, we can say that the name of a concept is the name of its currently preferred term. If we establish hierarchical relationships only among concepts and if we identify concepts with unique and meaningless numbers, we do not have to fear the constant change of terms and their meanings.

10) cf. Maniez, 1988, p. 220; cited by Nelson et al., 2001, section 4.1.1

2.3 Relationships of a Thesaurus

2.3.1 Equivalence Relationship

The relationship between entry and preferred terms is normally called equivalence relationship. Although the term equivalence seems to be appropriate, because the preferred term and its entry terms belong to the same concept, some critics complain about this term since the terms of a concept are in fact not equivalent.

Consider the example of Accounting and Bookkeeping. Although the developers of our minithesaurus agreed that both terms belong to the same concept it is very likely that this agreement is considered totally unacceptable by professionals of finance. And it is not only them who might complain about the claimed equivalence of these terms but also the linguistic faction who will give sophisticated reasons why these terms are far away from being equivalent. That's why it is very important to clarify for whom the thesaurus is meant, i.e. to define the user group, and which kind of documents will be indexed by means of the thesaurus.

Everybody who has ever developed a thesaurus can tell a story about the head-cracking discussions among professionals on the grouping of terms. In fact what we call an equivalence relationship can be decomposed into various relationships of different nature, among them relationships between

- colloquial and common terms
- common terms and foreign terms
- different spellings
- different grammatical forms
- inverted and not inverted forms
- abbreviations and long forms

The true equivalence relationship - linguistically spoken the *synonymy* - is in fact very rare and can maybe only found between spelling variants. Against the background of this discussion some people prefer the term "substitutionary equivalence" [11] to equivalence. If the use of this term prevents heated arguments about the grouping of term can be however doubted.

Here it is time for a general remark. Many discussions on thesauri can be considered irrelevant because they focus on aspects that are far beyond the scope of a thesaurus - like the linguistical ones we have mentioned above. There is no doubt that entry terms have to refer to their common preferred term and there is also no doubt that this reference has to be given a name in order to facilitate a discussion on this. But for the application of a thesaurus it is for instance not important to discuss which precise kind of relationship between Electronic

11) Nelson et al., 2001, section 4.2

banking and Cyberbanking is indicated by the UF code between them (by the way: it is the relationship between a colloquial and a common term). Entry terms merely serve the purpose to lead the user to the appropriate descriptor and although there might be some entry terms for a preferred term, there is only one preferred term for a concept.

2.3.2 Hierarchical Relationship

A thesaurus differs from a subject heading list insofar as it arranges its vocabulary in a hierarchical way, i.e. developers of a thesaurus try to clarify which terms are subordinated to other terms. This effort does not only satisfy intellectual needs but it serves also a real purpose. First the hierarchical level of a term that is assigned to a particular document reflects to what extent a topic is discussed in a document, i.e. in detail or only superficial. Second a strict hierarchy allows the user to search with a descriptor and all its subordinated terms (Down search) respectively a descriptor and all its superordinate terms (Up search). On principle there is no restriction in terms of the number of hierarchies a thesaurus might have, but one should always bear the ease of use in mind. Hierarchical relationships are as equivalence relationships expressed reciprocally, usually by means of the relationship indicators broader term (BT) and narrower term (NT). Consider the following example [12]:

Achievement
NT Academic Achievement
NT Black Achievement
[...]

Academic Achievement
BT Achievement

Black Achievement
BT Achievement

[...]

It is crucial to understand that hierarchical relationships exist only among preferred terms.

2.3.3 Kinds of the Hierarchical Relationship

Like equivalence relationships also hierarchical relationships can be categorised into different kinds. Consider the following fictious example:

College
NT Fachhochschule Hannover
NT Saxion Hogeschool IJselland
NT College of Arts
NT Student Council
NT College Administration

If we analyze this example carefully we will see that the five narrower terms belong to three

[12] cf. thesaurus of the Educational Resources Information Center; ERIC 2003

different kinds of hierarchical relationships:

Fachhochschule Hannover and Saxion Hogeschool IJselland are both concrete examples for the general term of a college; by borrowing terms of the world of object-oriented programming we can say that Fachhochschule Hannover and Saxion Hogeschool IJselland are instances of (the class) College. Accordingly we call this relationship the instance relationship.

College of Arts is a special kind of college. Such an IS-A relationship is professionally called a generic relationship. Z39.19 proposes a test by means of that one can check if a relationship is generic [13]. Applied to our case we have to check if ALL colleges of arts are colleges and SOME colleges are colleges of arts. Obviously our example passes this so-called all-and-some test.

Student Council and College Administration are both parts of a college. We can check such a whole-part relationship usally by whispering silently "College HAS A student council" and "College HAS A college administration".

We complained above about some annoying discussions regarding the development of thesauri that could be avoided since they are only of linguistic interest but irrelevant for the application of a thesaurus. In fact the majority of thesauri neglects the precise kind of a hierarchical relationship by just using BT and NT relationship indicators. However, sometimes it can be useful to use special relationship indicators, for instance in order to improve the readability of a long list of narrower terms. Z39.19 proposes BTP/NTP for the instance relationship, BTG/NTG for the generic relationship and BTP/NTP for the whole-part relationship [14]. Applied to our example we get

College
NTI Fachhochschule Hannover
NTI Saxion Hogeschool IJselland
NTG College of Arts
NTP Student Council
NTP College Administration

Note that the instance, generic and whole-part relationships are all hierarchical relationships. We can say ("some - all") that between these three relationships and the hierarchical relationship exists a generic relationship. This will be relevant in the course of this dissertation.

2.3.4 The Odd One Out: Cross-Referencing Compound Terms

Thus far we have not discussed that in order to represent some meanings we have to make use of more than one term. Such compound or multiword terms representing one unit of thought are also called *lexemes*. One example of a compound term is

Press conference

[13] cf. ANSI/NISO Z.39.19, 1993, p. 17

[14] cf. ANSI/NISO Z.39.19, 1993, pp. 17-18

On principle the developers of a thesaurus have two possibilities to deal with compound terms. Either they declare them descriptors apart from the single words the compound term consists of, or they consider them entry terms that have to be represented by the combination of single-word descriptors. The former one is also referred to as precoordination, the latter one as postcoordination, i.e. either the thesaurus co-ordinates the terms (pre) or the users have to do this (post).

Z39.19 provides several guidelines how to decide whether or not a compound term should be given descriptor status [15]. Among the various factors that should be considered is the question if the single words that the compound term consists of are within the scope of the thesaurus. In our case: if either press or conference are beyond the scope, the compound term press conference has to become a descriptor. Provided that it is sensible to assign both Press and Conference descriptor status we could represent the concept of a press conference by combining press *and* conference. In order to guide the user we would include press conference as an entry term in the thesaurus that refers to the two single-word descriptors:

Press conference
USE Press AND Conference

Sometimes abbreviated as

Press conference
USE+ Press, Conference

Note that this is something new: thus far we only have discussed relationships between at most two terms, also called binary relationships. Now we deal with a relationship among three terms. Unfortunately this will cause some troubles when it comes to the RDF representation. For consistency we express this relationship reciprocally:

Press
UF+ Press conference

Conference
UF+ Press conference

We use UF+ instead of UF in order to avoid the interpretation that Press is equivalent to Press conference or Conference to Press conference respectively. Since UF+ refers to the entry term rather than the combination of the single-word descriptors we deal with a binary relationship again [16]. Note that we have discussed intentionally a rather simple example in that the compound term equals the concatenation of two single-word descriptors:

Press conference = Press + Conference

However, this is not always the case. Consider the following example:

[15] cf. ANSI/NISO Z.39.19, 1993, p. 11

[16] ANSI/NISO Z.39.19, 1993, p. 16 explains UF+ as 'used for ... and ...'; this explanation differs from the meaning of the German counterpart KB (Kombinationsbegriff; Burkart, 1997, p. 175) that refers to the entry term and not to the term this descriptor can be combined with. However, we use UF+ in the meaning of KB.

Bug
USE Programme AND Malfunction

Programme
UF+ Bug

Malfunction
UF+ Bug

In that case the reciprocal UF+ entries are of little help since the user has no idea with which term Programme or Malfunction respectively has to be combined in order to represent a bug. Preferable would be the entries

Programme
UsedInCombinationWith Malfunction For Bug

Malfunction
UsedInCombinationWith Programme For Bug[17]

Although this contributes to the ease of use of the thesaurus it is insofar more complicated as we have to deal with another non-binary relationship. It can be doubted that the reciprocal entries are helpful in case of descriptor combinations. Is it important to know for a user who looks up the term Programme that this descriptor is used in combination with Malfunction in order to represent Bug? Obviously the developers of the ERIC thesaurus have denied this question; the entry term Achievement prediction refers to the combination Achievement and Prediction but neither Achievement nor Prediction refers to the entry term Achievement prediction [18]. We will bear this obviously not-too-harmful omission in mind.

2.3.5 The Associative Relationship

The associative relationship is normally defined ex negativo: every relationship that is neither equivalent nor hierarchical nor referencing compound terms and single-word descriptors is considered associative [19]. As often this definition is mentioned as useless it is. A better idea is to explain which purpose the associative relationship serves: it is all about user guidance. The more extensive a thesaurus is the more likely it is that a user does not find an appropriate descriptor for what he is thinking of. By specifying associative relationships, normally coded RT for related term, the developers of a thesaurus remind the user of the existence of other descriptors that might be more appropriate. Unfortunately this approach of trying to think of what the user might think of often leads to an overuse of the associative relationship. Consider the following example [20]:

17) cf. Burkart, 1997, p. 175
18) cf. ERIC, 2003
19) cf. DIN 1463-1, 1987, p. 10
20) cf. ERIC, 2003

Sample Descriptor Display

PREDICTION
SN Process or act of foretelling future events, conditions,
outcomes, or trends on the basis of current information
NT Employment Projections
NT Enrollment Projections
NT Grade Prediction
RT Chaos Theory
RT Decision Support Systems
RT Delphi Technique
RT Environmental Scanning
RT Estimation (Mathematics)
RT Expectation
RT Futures (of Society)
RT Long Range Planning
RT Markov Processes
RT Predictive Measurement
RT Probability
RT Regression (Statistics)
RT Reliability
RT Risk
RT Self Fulfilling Prophecies
RT Social Indicators
RT Strategic Planning
UF Forecast

Note that as a matter of course associative relationships exist only among descriptors. SN is an abbreviation for scope note that explains the meaning of the term. We will discuss it in detail later (see section 2.4.3).

Developers of a thesaurus should always bear in mind that the indication of related terms is an additional means to clarify the meaning of a descriptor. Often the specified hierarchical relationships suffice in order to understand what is meant by a term. Other means to facilitate the use of a thesaurus are described in the next section.

2.4 Annotating a Descriptor

For maintenace and usability reasons both terms and concepts are normally annotated. We describe the common annotations one after another.

2.4.1 History Note

A Thesaurus tries to control the vocabulary used by a certain group of people, normally belonging to a particular field of profession. Although this attempt might be successful the language that is used by the professionals changes in the course of time. New terms have to be added, some terms that are not used anymore have to be removed. Besides it is possible that some entry terms become preferred terms and vice versa. We can consider such changes a term is subjected to in the course of time the history of the term. It is strongly recommended to record the history of a term in order to facilitate the maintencance of the thesaurus. This is the purpose of the history note (usually coded HN).

In former times it was crucial for the user of a thesaurus to check carefully in which period of time a particular term was valid and to know the terms that precede and succeed. This complicated the search for documents over a large period of time significantly since it is very likely that different terms were used in say the last twenty years in order to describe the same concept. Thus users were forced to use different descriptors depending on the period of time they were searching in although they were searching for the same thing. Nowadays this is often not a big problem anymore since modern computer applications for thesauri are able to reindex documents, i.e. to substitute old terms with new ones automatically. But as a matter of course this is also only possible because of the carefully recorded history of a term. The example shows a typical history note [21]:

> FEDERAL REGION IX
>
> HN Prior to June 1982 this concept was indexed to WESTERN REGION
> DA June 7, 1982

Although the date of addition (above abbreviated with DA) and the date of deletion (if applicable) of a term are normally recorded separately from the history note we may consider them part of the history note since they provide the most important information when it comes to the term's history: its validity.

2.4.2 Source

Creators of a thesaurus consult various sources in order to find out which vocabulary is used by their clients: articles of professional newspapers and magazines, handbooks, interviews with the professionals, existing thesauri, classifications and subject heading lists of related fields of profession. Especially in the course of creation of a thesaurus it is crucial to clarify which terms are describing the same concept. Normally the creators of a thesauri face the difficult situation that the use of the vocabulary even within a rather small group of experts is far away from being standardised. Some professionals may use different terms for the same thing or - even worse - the same term for different things. That is why it is important to record in which source a particular term was found, in order to enable future users to check the context in that this term was originally used. It might turn out that its meaning was misunderstood either by the creators of the thesaurus or by the writer himself. In the light of its importance it is a pity that many thesauri neglect the source of a term.

2.4.3 Scope Note and Definition

In many cases it is very important to support the user of a thesaurus in finding the appropriate term to search or describe a document. Although the term itself should be as self-explanatory as possible very often its meaning in the context of the thesaurus and its proper application

[21] cf. International Energy Subject Thesaurus; IEST, 1990

have to be explained. Put another way, it should be clarified what the scope of a particular term is. That is why many terms are explained by means of a scope note (SN). Consider the following example [22]:

> AUTHORS
> SN Use for biographical discussion of particular authors and their
> works. For works chiefly discussing an author's work, with little
> biographical component, use Literature or its subordinate headings

Note that the scope note of a term differs from its definition. While a scope note attempts to clarify the proper use of a term by explaining which kind of content should be described by means of this term a definition clarifies the meaning of this term in a particular field of profession. Thus the former one is only valid within the thesaurus while the latter one should be also valid outside the thesaurus. Definitions are normally found in standard descriptions [23].

It is not always easy to distinguish between a scope note and a definition since a definition also contributes to the proper application of a term. The following example illustrates this difficulty [24]:

> ACTIVISM
> SN Movements and procedures designed to force changes in rules and
> practices or to hasten social change

This scope note seems to defining the term in the context of social science rather than explaining which kind of content should be described with it. But after all both scope note and definiton serve the same purpose of clarifying the meaning of a term. Its precise distinction is subject of a rather theoretical discussion that is far beyond the scope of this dissertation.

2.4.4 Category

The more terms a thesaurus consists of the more difficult it is to overview its content. There are many means to improve the readibility and in this way the ease of use of a thesaurus; one of them is the grouping of descriptors that are somehow associated with each other. It is crucial to understand that this grouping does not reflect any hierarchical relationships among descriptors; in contrast to the categories of a classification descriptors are rather grouped by sub-fields or mere topics of a certain field of profession. Consider the following sample category and their subcategories [25]:

[22] cf. APAIS, 2002

[23] cf. Burkart, 1997, p. 176

[24] cf. ERIC, 2003

[25] cf. Thesaurus of Engineering and Scientific Terms; TEST, 1967

0100 Aeronautics

0103 Aircraft
Aerial rudders
Aerodynamic brakes
Aerodynamic configurations
[...]

0104 Aircraft Flight Instrumentation
Aircraft equipment
Aircraft instruments
Airspeed indicators
[...]

In this example each category is assigned a unique number. Note that 0103 Aircraft and 0104 Aircraft Flight Instrumentation are subcategories of 0100 Aeronautics. Although some thesauri might support subcategories the descriptors are always assigned to categories of the lowest hierarchical level, i.e. in case of the TEST Thesaurus there may be no descriptor belonging to the supercategory 0100 Aeronautics.

The assignment of descriptors to categories may be not mistaken for the categorisation as defined by a classification. The purpose of the categorisation of descriptors is merely to provide an additional display of descriptors that might facilitate the access to the descriptors. To what extent this purpose is fulfilled can be doubted and therefore should be subject of another examination.

PART 3
XML and Thesauri

3.1 What is XML?

It is easier to start with the counterquestion: What is XML (eXtensible Markup Language) [26] not? It is not what it pretends to be: a markup language. Rather it is a standard for developing markup languages. Thus we remain with having to clarify what a markup language is: A markup language is the crucial means to create self-describing data. - For decades we were facing the problem that the data could not be separated in a sensible way from the application that stores the data. Suppose we have a database storing data about a company, e.g. its customers, employees, transactions and goods. Provided that we have learned our lesson we will backup our data on a regular basis, i.e. exporting it from the application to for example a tape. If we removed the application the data would be totally meaningless not only for the human reader but also for all other kinds of applications. The reason for this is that every application exports its data in its own way; put another way: every application has its own data format. Critics might argue that the majority of applications is capable to export their data in the so-called CSV format (CSV stands for comma-separated values). This format ensures that the characters are encoded in a human-readable way and therefore prevents us from having to scan files looking like a mixture of hieroglyphs and runes. Usually this export format is chosen for data in tabular form that are very popular owing to the success of relational databases. Let us have a closer look at a sample entry of a CSV file:

```
Paine, Thomas, Common Sense, 1776
```

Obviously the application that has produced this file stores data about books. Thomas Paine is the author of the pamphlet "Common Sense" that was published in 1776. So what? Data export and therefore exchange does not seem to be a problem at all. But we may not neglect the influence of our general knowledge in order to interpret this file properly. Suppose we were confronted with another CSV file:

```
0004184283, Y, +491183852055, 030501, 4
```

Okay, we do not have to give up on the spot. Maybe the first value is a kind of unique identifier, Y is very likely to stand for Yes (but yes, what?), +49 is the country code of Germany, thus we might draw the conclusion that this is a telephone number. The second last value could be a date, but what is 4? The number of children, a status, the number of purchases? The problem is that the meaning of the data was lost during the export. It is only the application that exported this file that "knows" that the rightmost value stands for how

26) cf. XML, 2000

often this customer has been called.

Markup languages contribute to overcome this problem by providing a means for storing both data and their descriptions in the same file. As a matter of course we somehow will have to distinguish the data and its description if we plan to store both in a single file. XML has reserved the angle brackets for this purpose, but since it is a standard and not a language, XML itself does not provide any data labels. The descriptions are part of a concrete markup language. A valid XML representation of our last example might look like this:

Sample XML Document

```
<customer>
  <id>0004184283</id>
  <approved>Y</approved>
  <telephone>+491183852055</telephone>
  <lastCall>030501</lastcall>
  <totalCalls>4<totalCalls>
</customer>
```

Note that this markup language comprises only six data labels or *elements*.

This is much better than a CVS file - not only for a human reader but also for an application: the good thing with XML is that every application that is able to process XML files can process all files whose markup language was developed according to XML.

Fortunately the XML standard does not request too much, but its few rules have to be strictly obeyed. The majority of them can be derived from our example:

- XML is case-sensitive, i.e. lastCall differs from lastcall.
- Every starting tag like <id> has to have a closing tag like </id>.
- Elements have to be properly nested.

As this introduction shall only prepare the reader for the following parts of this dissertation we are not going to discuss XML in detail but only the parts that are relevant against the background of thesauri representation. If required we will introduce more features of XML.

3.2 Zthes: A Markup Language for Thesauri

XML is a typical example of the hierarchical approach to organise data and bearing in mind that a thesaurus is nothing but a hierarchically arranged vocabulary, XML seems to be perfectly appropriate in order to represent thesauri.

In fact we are not the first people thinking of this possibility. Zthes [27] is an XML markup language that is meant for describing thesauri [28]. Among the thesauri that are described by means of this markup language there is the thesaurus used by the Australian Public Affairs Information Service (APAIS) [29], part of the Australian National Library. APAIS indexes

[27] cf. Zthes, 2001

[28] Note that Zthes defines merely an abstract model for representing and searching thesauri. Zthes is mainly meant for being implemented by the Z39.50 (cf. ANSI/NISO Z.39.50, 1995) protocol; however the implementation by other protocols and data formats is obvious.

[29] cf. APAIS, 2002

articles in the humanities and social science published in Australia.

Before we have a look at a sample descriptor represented by means of the Zthes markup language let us consider how this descriptor might appear in the print display in order to discuss which meaning we are going to encode in XML.

Marxism
UF Communism
UF Historical Materialism
BT Political science
RT Socialism

Owing to our cultural background, we do not have problems to understand the meaning of this display since we are used to the method of indenting that indicates the subordination of one term to another term. But let us explicitly write down which statements we derive from this display:

Marxism UF Communism
Marxism UF Historical materialism
Marxism BT Political science
Marxism RT Socialism

Without having to look up the referred terms we can also derive the following statements:

Communism USE Marxism
Historical materialism USE Marxism
Political science NT Marxism
Socialism RT Marxism

Let us now have a look how these statements were encoded in the APAIS thesaurus by means of the Zthes markup language [30]:

Descriptor Representation in XML/Zthes

```
<thes>
  [...]
  <term>
    <termId>R0808</termId>
    <termName>Marxism</termName>
    <termType>PT</termType>
    <termNote>Use for political theories, parties
    and movements grounded in the ideas of Marx,
    aiming towards a society embodying common
    ownership of the means of production,
    distribution and exchange. Class here Marxist
    theories advocating the revolutionary overthrow
    of the state</termNote>
    <termCreatedDate>1981</termCreatedDate>
    <termModifiedDate>9/04/2002</termModifiedDate>
    <relation>
      <relationType>UF</relationType>
      <termId>N0144</termId>
      <termName>Communism</termName>
      <termType>ND</termType>
```

[30] cf. APAIS, 2002; Zthes distinguishes between preferred terms (PT) and non-descriptors (ND)

```
      </relation>
      <relation>
        <relationType>UF</relationType>
        <termId>N0393</termId>
        <termName>Historical materialism</termName>
        <termType>ND</termType>
      </relation>
      <relation>
        <relationType>BT</relationType>
        <termId>R0988</termId>
        <termName>Political science</termName>
        <termType>PT</termType>
      </relation>
      <relation>
        <relationType>RT</relationType>
        <termId>R1185</termId>
        <termName>Socialism</termName>
        <termType>PT</termType>
      </relation>
    </term>
    <term>
      [...]
    </term>
  </thes>
```

At first glance we could have the impression that the statements mentioned above are again encoded by indenting. But even if this example was written in a single line it would remain a perfectly valid XML document. In fact the statements are conveyed with the help of nested elements. If we considered an element regardless of its context it would immediately loose its meaning:

```
<termName>Historical materialism</termName>
```

But within the context the statement 'Marxism UF Historical materialism' can be derived from this XML representation since termName is a subelement of relation and relation is a subelement of term. The type of relation is indicated by the element relationType residing on the same hierarchical level like termName. In fact we always have to interpret the containment of an element against the background of all elements enclosing it - up to the uppermost level. This approach of encoding the meaning by means of nesting items is typical for the hierarchical approach of storing data.

Since XML trees can be despite of indenting very difficult to read and only clumsily described in natural language we introduce a useful tool describing the precise position of an element unambiguously: the XML Path Language (XPath) [31]. XPath is not a language but it is used by XML extension languages in order to locate elements within an XML tree. The fact that termName is a subelement of relation is a subelement of term is a subelement of thes is expressed in XPath as follows:

```
/thes/term/relation/termName
```

Note that the leading slash indicates that this path is absolute, i.e. there is no additional hierarchical level above thes. Since this XPath expression would select all termName elements that reside on this hierarchical level we have to be more precise by adding the

[31] cf. XPATH, 1999

following information:

- termName contains 'Historical materialism'
- relation has a subelement relationType that contains 'UF'
- term has a subelement termName that contains 'Marxism'

Fortunately XPath supports such a request:

Sample XPath Expression

```
/thes/term[descendant::termName='Marxism']/relation[
descendant::relationType='UF']/termName[node()='Historical
materialism']
```

Even without having explained XPath in detail, the reader should be able to understand the meaning of this expression.

3.3 Adding a Grammar to XML: DTD and XML Schema

Although the XML representation we have discussed makes sense sceptics might ask by which means it is ensured that elements are not nested in a senseless way. Some senseless arrangements could be for example

```
/thes/relation
/thes/term/thes
/thes/termName
```

Unfortunately such arrangements would be still perfectly valid XML documents. The problem is that native XML does not provide any means to disallow certain nestings that would be senseless. But in our case it would be very helpful if we could add some constraints such as:

- There may be only one thes element that has to appear at the uppermost level.
- At the second hierarchical level there may occur only term elements, as many as required but at least one.
- The term element has to contain (in this sequence) exactly one termId, one termName, zero or one termType, zero or one termNote and as many relation elements as required.

What we are searching for is a grammar for XML. There are two possibilities to add a grammar to XML documents: either by means of a Document Type Definition (DTD) [32] or by means of an XML Schema [33]. The latter is the modern one and is more and more substituting the former. We are not going to discuss these methods in detail. It is only important to understand that the constraints we have started to itemize above can be expressed with the help of a DTD or an XML Schema. It is also crucial to know that this is normally done not within the XML document but in a separate file the XML document has to refer to. An XML application that does not only check if an XML document complies with

[32] cf. XML, 2000

[33] cf. XSchema, 2001

the XML standard but also checks if it complies with the grammar defined either in a DTD or XML Schema is said to validate the document. Note that such a validation requires always the presence of the DTD or XML Schema respectively and that by far not all XML applications are able to validate an XML document. In fact the Zthes markup language for Thesauri does define such a grammar by means of a DTD [34].

3.4 Shortcomings of XML

Although the Zthes markup language obviously serves its purpose we may not forget that this markup language is only one among theoretically countless possibilites to represent thesauri in XML. Consider our previous statement again:

Marxism UF 'Historical Materialism'

and its representation according to Zthes:

```
<thes>
  <term>
    <termName>Marxism</termName>
    <relation>
      <relationType>UF</relationType>
      <termName>Historical materialism</termName>
    </relation>
  </term>
</thes>
```

We could represent the same statement as well as

```
<thes>
  <term>
    <termName>Marxism</termName>
    <usedFor>
      <termName>Historical materialism</termName>
    </usedFor>
  </term>
</thes>
```

or maybe as

```
<thes>
  <term>
    <termName>Marxism</termName>
    <relation type="UF">
      <termName>Historical materialism</termName>
    </relation>
  </term>
</thes>
```

or maybe as

```
<thes>
  <term name="Marxism">
    <relation type="UF" name="Historical materialism"/>
```

[34] see Appendix 1

```
  </term>
</thes>.
```

All representation are valid XML and encode exactly the same statement. The problem is that each document produces a different XML tree and thus we have to apply different methods in order to derive the same statement [35]. Sticking to our sample representation, we would have to query the statement 'Marxism UF Historical materialism' in four different ways by XPath:

Sample XPath Expressions querying the same Statement

```
/thes/term[descendant::termName='Marxism']/relation[
descendant::relationType='UF']/termName[node()=
'Historical materialism']

/thes/term[descendant::termName='Marxism']/usedFor/
termName[node()='Historical materialism']

/thes/term[descendant::termName='Marxism']/relation[
attribute::type='UF']/termName[node()=
'Historical materialism']

/thes/term[attribute::name='Marxism']/relation[
attribute::type='UF' and attribute::name=
'Historical materialism']
```

Of course, provided that we have the corresponding DTD or XML Schema we would exactly know how to query this statement. As we want to develop a standard for all thesauri we have to pose the question how likely it is to define a DTD or XML Schema that is applicable for all thesauri. Although there are several thesauri reprensented by means of the Zthes markup language this is unfortunately very unlikely. The reason for this is that it depends on the particular field of profession and the wishes of the users how a thesaurus looks like. The DTD of the well-known Medical Subject Headings (MeSH) thesaurus [36] of the US American National Library of Medicine for example comprises 83 elements in contrast to the Zthes DTD that only defines 15 elements. This would not be a big problem if all thesaurus developers agreed on a minimum set of elements that could be extended. But unfortunately DTD are not extensible since the mere addition of a single element results in a different grammar and thus requires different queries. In addition, since an XML document does not have to refer to a DTD or an XML Schema and many XML applications do not support the validation of an XML document it can not be guaranteed that the constraints of the grammar are always obeyed. The crucial point is that XML - although it provides an unambiguous syntax - does not have an inherent unambiguous grammar.

[35] cf. Berners-Lee, 1998, section: The XML Graph

[36] cf. MeSH, 2003

PART 4

RDF and Thesauri

4.1 What is RDF?

Suppose you want to search the Web for a college in Great Britain offering courses of study in life sciences. You do so by using your favourite search engine and the keywords 'college', 'Great Britain' and 'life sciences'. Consider the following results:

> *Possible Search Results for 'college', 'Great Britain' and 'life sciences'*
>
> `penfriends.com:` Hi, I am searching for a pen friend in Great Britain. I am a 18-year-old student at a college in Greece. My hobby is reading, especially books about Astronomy and Life Sciences.
>
> `biotech.com:` Among the current vacancies are: lab assistant in our branch in Great Britain. Prerequisites: You hold a degree in Life Sciences - preferable obtained at a college.
>
> `news.com:` During a visit of a research institution in South Wales the British prime minister said that the government was well aware of the increasing importance of Life Sciences in Great Britain. Among his visits in this region was also a recently founded college.

One could say that the search engine misunderstood [37] you. But this would be an exaggeration, in fact it did not understand you at all! The problem is that there is no way to say that you are searching for 'an institution that is a college located in Great Britain offering Life Sciences'. Without analyzing these examples in detail it is obvious that the information published on the Web can be only understood by human beings, not by machines. Put another way, although the information is digital and therefore machine-processable it is not machine-understandable. Scientists all over the world are busy with developing methods how information can be stored in a machine-understandable way. In terms of the Web the aim is to develop a Semantic Web that understands the information it provides. Such a Semantic Web might enable us in the future to query the Web more successfully than nowadays. Fortunately today there are several methods available that serve this purpose, among them the Resource Description Framework (RDF) [38], a framework that is maintained by the World Wide Web Consortium (W3C) [39].

RDF is considered a rather simple (or light-weight) language to make information understandable. But this simplicity could contribute significantly to a successful realisation of

[37] We are well aware of the fact, that machines never understand anything; for simplicity we will not quote the verb understand being connected with machines although this would be more accurate.

[38] cf. RDF, 1999

[39] cf. W3C, 2003

the semantic web. In fact RDF's applicability is not confined to the description of digital information such as a (part of a) website that is accessible online. On principle by means of RDF everthing can be described that can be given a unique identifier, this may be a concrete item like a book, a dish of a restaurant or an employee of a company; or an abstract one like a descriptor of a thesaurus. Adopting the RDF terminology we say that RDF is meant for describing resources given an uniform resource identifier (URI) [40].

It is crucial to understand that RDF, although it plays an important role for the Web, is not a computer language, but a conceptual language. This means that the language itself does not provide any syntax for its representation. However, usually this is done in three ways: by means of nodes-and-arcs diagrams, by means of triples or by means of XML. In the course of the following sections we will mainly make use of the triple notation. Afterwards we will discuss the relationship of RDF and XML in detail.

RDF consists of two parts: RDF [41] and RDFS [42] ('S' stands for Schema). The former one is meant for describing resources, the latter one is meant for the description of the vocabulary that is used to describe resources. Applied to our introduction we can say that the statement 'college located in Great Britain' is a subject of RDF and the definition of the predicate 'located' is a subject of RDFS. It may be a bit confusing that RDFS uses also elements of RDF and that both parts together are referred to as RDF, but having understood the general distinction this should not be a big problem. Since at first we are going to describe the descriptive vocabulary of a thesaurus, e.g. narrowerTerm, usedFor, Concept etc., we will start with RDFS. Afterwards we will make use of RDF to represent a certain thesaurus by means of the previous defined descriptive vocabulary, e.g. 'Marxism usedFor Historical Materialism'.

4.2 Back in school: Subject, Predicate, Object

Since human beings encode statements by means of natural language, all approaches to make information machine-understandable rely heavily on the research of linguists. Let us consider a previous used example again:

Marxism UF Historical Materialism

In plain English we would express this as:

Marxism is used for Historical Materialism.

Remembering our first lessons in primary school, we recognize the grammar of this sentence:

Subject: Marxism
Predicate: is used for
Object: Historical Materialism

[40] cf. RFC 2396, 1998 and RFC 2141, 1997

[41] cf. RDF, 1999

[42] cf. RDFS, 2003

Although we make the developers of the APAIS thesaurus cry, it is grammatically spoken correct if we exchange Marxism and Historical Materialism:

> Historical Materialism is used for Marxism.
>
> Subject: Historical Materialism
> Predicate: is used for
> Object: Marxism

But it would be unreasonable if we declared 'is used for' a subject and Marxism a predicate:

> Subject: is used for
> Predicate: Marxism
> Object: Historical Materialism

Obviously it is in accordance with English grammar to exchange subject and object - although the meaning changes - but a predicate may become neither a subject nor an object. Thus we have two kind of things: Things that may be subject or object and things that may be a predicate. In RDF the former ones are called classes and the latter ones are called properties.

4.3 Classes and Properties of a Thesaurus

Before we can continue with RDF, we have to clarify which parts of a thesaurus are classes and which parts are properties. In part 2 we discussed theoretically the construction of a thesaurus. One crucial outcome of this discussion is that we have to distinguish between a term and a concept. There is no doubt that neither a term nor a concept may function as a predicate. Therefore we may consider both classes as defined by RDF rather than properties. Although we are not obliged to do so we will adopt a de-facto standard according to that classes always start with an uppercase letter whereas properties start with a lowercase letter.

Classes of a Thesaurus

- Concept
- Term

It might strike some readers as rather strange that we do not mention descriptor as a class of a thesaurus. But remember that this was also one outcome of the theoretical discussion that a term can easily change its status, i.e. an entry term can become a preferred term and vice versa. That is why we should not define two classes for these two kinds of terms. As a matter of course the status of a term has to be somehow reflected.

From part 2 we can derive the following list of properties

Properties of a Thesaurus

- source
- historyNote

- dateOfAddition
- dateOfDeletion
- scopeNote
- category
- definition
- relatedConcept
- usedFor/use
- broaderConcept/narrowerConcept
- broaderConceptGeneric/narrowerConceptGeneric
- broaderConceptPartitive/narrowerConceptPartitive
- broaderConceptInstance/narrowerConceptInstance
- type
- comprisesTerm/belongsToConcept
- label

The last three properties were not mentioned before and therefore have to be explained. Type is used in order to indicate whether a term is an entry or preferred term [43], comprisesTerm/belongsToConcept in order to link a term to a concept (and vice versa) and label for the human-readable name of a term.

Considering the first seven properties we recognize the necessity of an additional class for plain text like 'Moral and ethical considerations in the life sciences field' [44] in case of scopeNote or '2003-05-05' in case of dateOfAddition. Fortunately RDFS itself provides a class for this kind of data, called Literal.

4.4 Defining Classes and Properties in RDF

After having identified the Classes and Properties of a thesaurus we have to define them in a formal way in RDF. Formal definitions in RDF can be hard to read since normally terms from different vocabularies are involved. In order to prevent misunderstandings from the start we introduce a useful tool by which we can indicate to which vocabulary a term belongs to. For each vocabulary we define an abbreviation that is added to the appropriate term as a prefix, separated from the term by a colon. The widely used prefixes for RDF and RDFS are - what a surprise - rdf and rdfs. For the vocabulary that we are about to define we will use the prefix thes. Having done this, we can define some sample classes and properties. rdf:type is reserved for this purpose:

Sample Class and Property Definitions in RDF

```
thes:Concept rdf:type rdf:Class .
thes:Term rdf:type rdf:Class .
thes:relatedConcept rdf:type rdf:Property .
thes:source rdf:type rdf:Property .
```

[43] type could be also used to indicate whether a descriptor may be used for indexing or merly serves as a categoriser like the MeSH (cf. MeSH, 2003) non-descriptors

[44] APAIS, 2002, Scope note of Bioethics

4.5 Adding Restrictions: Domains and Ranges

By distinguishing between classes and properties we did an important step in order to develop a grammar of thesauri. Suppose we stop the development of our grammar at this step a computer application applying our grammar could create the following statements:

Valid but Senseless RDF Statements

```
Concept thes:dateOfAddtion Term .
e.g. 4711 thes:dateOfAddition wine .

Term thes:narrowerTermGeneric Literal .
e.g. liver thes:narroverTermGeneric 'serve with beans' .

Literal thes:usedFor Concept .
e.g. 'till 1987 indexed with "Insanity"' thes:usedFor
0815 .
```

The statements comply with our grammar, but obviously they are not in accordance with our common sense. We somehow have to define that certain properties may only link certain classes; and we may not neglect that the order of the elements of a statement matters, i.e. although a class may on principle function as both an object and a subject, it can be sensible to restrict a class to be either a subject or an object - depending on the property. RDF provides the means to apply such restrictions: domain and range. The domain of a property indicates which class may precede it, i.e. may function as the subject, and range indicates which class may succeed it, i.e. may function as an object of this property. Consider a sample domain and range specification in RDF:

Sample Domain and Range Definition

```
thes:comprisesTerm rdfs:domain thes:Term .
thes:comprisesTerm rdfs:range thes:Concept .
```

This means that the property comprisesTerm may only link the class Term with the class Concept - and only in this order. Accordingly we have to define all properties. The following table itemizes the properties of our thesaurus:

Properties and their domains and ranges

Property	**domain:**	**range:**
source	Concept	Literal
historyNote	Concept	Literal
dateOfAddition	Concept	Literal
dateOfDeletion	Concept	Literal
source	Term	Literal
historyNote	Term	Literal
dateOfAddition	Term	Literal
dateOfDeletion	Term	Literal
label	Term	Literal

Property	domain:	range:
type	Term	Literal
category	Term	Literal
definition	Term	Literal
belongsToConcept	Term	Concept
comprisesTerm	Concept	Term
relatedConcept	Concept	Concept
broaderConcept	Concept	Concept
narrowerConcept	Concept	Concept
broaderConceptGeneric	Concept	Concept
narrowerConceptGeneric	Concept	Concept
broaderConceptPartitive	Concept	Concept
narroverConceptPartitive	Concept	Concept
broaderConceptInstance	Concept	Concept
narrowerConceptInstance	Concept	Concept
usedFor	Term	Term
use	Term	Term

4.6 Subclasses

It strikes that according to this itemization Term and Concept share some properties, namely source, historyNote, dateOfAddition, dateOfDeletion. As RDF offers the possibility to define zero, one or many domains and ranges per property, this does not seem to be a big problem. But according to the RDF semantics, the specification of more than one domain per property means that any resource having this property has to be an instance of *all* classes specified as the domains. The same applies to the specification of ranges [45]. Since a resource will never be an instance of both a concept *and* a term we may not specify two domains per property. In order to solve this problem we borrow a well-known method of object-oriented programming, that is fortunately supported by RDF: subclasses. We introduce a superclass called ThesaurusItem and define Concept and Term as subclasses of this new class. In RDF this is done by means of subClassOf:

Definition of Subclasses

```
thes:Term rdfs:subClassOf thes:ThesaurusItem .
thes:Concept rdfs:subClassOf thes:ThesaurusItem .
```

Having done this, we can specify ThesaurusItem as the domain of all properties that are shared by Term and Concept. As a matter of course the range remains the same:

Domain and Range Defintion of a Superclass' Property

```
thes:source rdfs:domain thes:ThesaurusItem .
```

[45] cf. RDF Primer, 2003, section 5.2

```
thes:source rdfs:range rdfs:Literal .
```

The same applies to historyNote, dateOfAddition and dateOfDeletion.

It is crucial to understand that subClassOf is transitive, i.e. an instance of Term is also an instance of all super-classes of Term. If this was not the case subclassing would be of no use for us since we want to allow both Concept and Term to make use of the same properties. On principle a class may have more than two direct super-classes. However, this is not required in our case.

4.7 Subproperties

In section 2.3.3 we discussed the different kinds of hierarchical relationships: generic, partitive and instance. We also stated that between these kinds of relationships and the general hierarchical relationship exists a generic relationship. That means that for example broaderConceptGeneric IS-A broaderConcept. In RDFS we say that broaderConceptGeneric, broaderConceptPartitive and broaderConceptInstance are sub-properties of broaderConcept. Accordingly we define the relationship between narrowerConcept and its subproperties. Formally this is done by means of rdfs:subPropertyOf:

Sample Definition of a Subproperty

```
thes:broaderConceptGeneric rdfs:subPropertyOf
thes:broaderConcept .
```

The meaning of this statement is that broaderConceptGeneric is implicitly considered a broaderConcept, i.e. if we link two resources by means of broaderConceptGeneric we do not have to link them by means of broaderConcept since this relationship is implied.

4.8 The Odd One Out: The Follow-Up

In the course of the theoretical discussion we found out that the relationship between compound terms and single-word descriptors is an extraordinary one. The crucial point is that this relationship links more than two classes; RDF however intrinsically only supports binary relationships in the form 'subject predicate object'. RDF proposes some methods to deal with relationships of a higher arity than two. But unfortunately none of these methods is capable to express the meaning that a compound term refers to the combination (or intersection) of two single-word terms [46]. In fact this is a well-known shortcoming of RDF that justifies the existence of richer schema languages like DAML+OIL [47], that provide - among other capabilities - also the expression of intersections.

For the first time we get to know the limitations of RDF. Do we have to consider RDF now inappropriate for the representation of thesauri? Well, even before people started to think

[46] cf. RDF Primer, 2003, section 5.5

[47] cf. DAML+OIL, 2001

about computer-understandable definitions of thesauri, cross-references between single-word descriptors and compound terms have raised sophisticated discussions, i.e. we are not the first persons who have problems with this kind of relationship. Besides, RDF is still under development und the W3C is always grateful for proposals how the RDF data model can be extended and maybe one day it will be enriched by the ability to describe classes in terms of combinations. Thus, instead of giving up now, we may continue.

PART 5

An RDF Schema for Thesauri

5.1 Putting Everything Together

Thus far we have only discussed small parts of our RDF schema. In order to prevent us from losing track of things we should zoom out and sum up. The interested reader finds our RDF schema for thesauri in appendix 2.

5.2 CERES: Another RDF Schema for Thesauri

Unfortunately we may not pride ourselves on being the first persons who have succeeded in developing an RDF Schema for thesauri. What we have just done was also done by California Environmental Resources Evaluation System (CERES) cooperating with the United States Geological Survey Biological Resources Division (USGS/BRD) for the CERES/NBII Thesaurus Partnership Project [48]. We will discuss in section 5.4 why these institution have developed an RDF schema for thesauri. At first we will have a look on their result, that as a matter of course also influenced the development of our schema.

CERES' classes

- Term
- Category (subclass of Term)
- EntryTerm (subclass of Term)
- Descriptor (subclass of Term)

CERES' properties

Property	**domain:**	**range:**
status	Term	String [49]
source	Term	String
HN (history note)	Term	String
SN (scope note)	Descriptor	String
CN (cataloguer note)	Descriptor	String
UF	Descriptor	EntryTerm
USE	EntryTerm	Descriptor
CAT (category)	Descriptor	Category
Descriptor [50]	Category	Descriptor

[48] cf. CERES/NBII, 1998

[49] String is an outdated class; in the current RDFS specification Literal supercedes String.

Property	domain:	range:
TT	Descriptor	Descriptor
RT	Descriptor	Descriptor
NT	Descriptor	Descriptor
BT	Descriptor	Descriptor

The major difference between the CERES schema and ours is that CERES does not distinguish between a concept and a term. In section 2.2 we have given important reasons for this distinction. However, we are well aware of the fact that within the thesaurus community the discussion about this distinction is still going on. Although nobody questions the difference between a concept and a term it might depend on the particular case whether or not this distinction should be explicitly represented. Therefore one may consider the CERES schema as an example of the term-based approach and our schema as an example of the concept-based approach. After all it depends on which kind of thesaurus should be represented when it comes to the choice of the appropriate schema. Another important difference is that the CERES schema considers a category a class rather than a property of term. We refer to section 2.4.4 why this should not be done in a thesaurus. Besides it uses an additional property called TT (for top term) that specifies the broader term of the uppermost level. We consider this unnecessary since the top term can be derived from the hierarchy of broader terms; therefore it does not have to be specified explicitly. If you want to go into detail by checking the RDF Schema of CERES [51] bear in mind that this schema was developed several years ago. In that time RDF was still a draft; that is why you will find some outdated RDF(S) properties and classes. However, the basic data model has not changed.

5.3 Representing a sample Thesaurus by means of the RDF Schema

Thus far we have only discussed the development of our schema by using RDFS, i.e. we have described the descriptive vocabulary of a thesaurus. But we still owe the reader a sample RDF description of a concrete descriptor display by means of this schema. At first consider the print display [52]:

Print Display of a Descriptor

```
Concept [#C1]:
 Comprises Term: Activism [#T1] (preferred)
 Comprises Term: Militancy [#T2]
 Comprises Term: Political Protest [#T3]
 Broader Concept: Social Behaviour [#C2]
 Related Concept: Citizen Participation [#C3]
```

[50] Note that the label 'Descriptor' is used by CERES for both a Class and a Property

[51] cf. CERES/NBII, 1998 and appendix 3

[52] cf. ERIC, 2003; ERIC is a term-based thesaurus, we converted this display into a concept-based one for demonstration purposes.

Related Concept: Civil Disobedience [#C5]
Related Concept: Demonstrations (Civil) [#C6]
Related Concept: Dissent [#C7]
Related Concept: Lobbying [#C8]

We follow a general guideline to identify both concepts and terms not by means of their labels but by means of a meaningless identifier (above denoted by square brackets) whereby a leading C denotes a concept and a leading T denotes a term. By doing so we worsen the readability but please bear in mind that and RDF description is meant for machines not for human beings.

Representation of a Descriptor in RDF

```
eric:C1 rdf:type  thes:Concept .
eric:C1 thes:label  "Activism" .
eric:C1 thes:comprisesTerm eric:T1 .
eric:C1 thes:comprisesTerm eric:T2 .
eric:C1 thes:comprisesTerm eric:T3 .
eric:C1 thes:broaderConcept eric:C2 .
eric:C1 thes:relatedConcept eric:C3 .
eric:C1 thes:relatedConcept eric:C4 .
eric:C1 thes:relatedConcept eric:C5 .
eric:C1 thes:relatedConcept eric:C6 .
eric:C1 thes:relatedConcept eric:C7 .
eric:C1 thes:relatedConcept eric:C8 .

eric:T1 rdf:type  thes:Term .
eric:T1 thes:label  "Activism" .
eric:T1 thes:type "preferredTerm" .
eric:T1 thes:belongsToConcept eric:C1 .
eric:T1 thes:scopeNote "Movements and procedures designed
to force
changes in rules and practices or to hasten social
change" .
eric:T1 thes:usedFor eric:T2 .
eric:T1 thes:usedFor eric:T3 .

eric:T2 rdf:type  thes:Term .
eric:T2 thes:label  "Militancy" .
eric:T2 thes:type "entryTerm" .
eric:T2 thes:belongsToConcept eric:C1 .
eric:T2 thes:use eric:T1 .

eric:T3 rdf:type  thes:Term .
eric:T3 thes:label  "Political Protest" .
eric:T3 thes:type "entryTerm" .
eric:T3 thes:belongsToConcept eric:C1 .
eric:T3 thes:use eric:T1 .
```

5.4 Thesaurus Integration or Everything is a Resource

As stated earlier, everything that is described by means of RDF is referred to as resource and identified by means of an URI. But in fact, everything in RDF is a resource encompassing not only the things we describe but also the properties and the classes that are used for this description. In the course of this dissertation we used prefixes like rdf:, rdfs: or thes: in order to denote from which vocabulary a term originates. These prefixes are not only useful but absolutely required since it is probable that different vocabularies make use of the same term. For instance, both the Dublin Core [53] and the RDF vocabularies define an element called Description - each vocabulary for a different purpose; therefore we have to distinguish between DC:Description and RDF:Description. In fact, the prefixes are only shorthands; being processed by a machine they are substituted with URIs, for instance

eric:C1 rdf:type thes:Concept .

becomes

```
http://www.thesaurus.org/eric#C1
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.agnosis.de/rdfs/thesaurus#Concept .
```

whereby the full URI points to the location of the RDF description of the respective vocabulary. The hash character (#) is followed by the unique identifier of a particular element within an RDF description. This allows us not only to address a vocabulary but also a particular element of a vocabulary. Although the RDF: and RDFS: prefixes are widely used for the RDF and RDFS vocabularies you may use other prefixes since prefixes are only used within an RDF description. Some persons prefer S: to RDFS: for example. In the beginning of an RDF description you will always find the definition which prefix is used for which vocabulary. As simple this approach is as important are its consequences. In fact thanks to this approach we can overcome one of the major obstacles we have faced in the XML part. Suppose some thesaurus developers want to make use of our RDF schema. Unfortunately for the users of their particular thesaurus it is important to distinguish between different kinds of the equivalence relationship. Instead of only specifying usedFor, they are more precise by distinguishing among

usedForAbbreviation
usedForForeignTerm
usedForTrademark .

Since our RDF schema does not distinguish the different kinds of the equivalence relationship it seems at first glance as if they cannot make use of it. But having understood the resource approach of RDF, we know how the developers of the thesaurus in question achieve their aim without starting to write an RDF schema from the scratch. The usedForAbbreviation, usedForForeignTerm and usedForTrademark properties are all specializations of the usedFor property of our RDF schema, thus we may consider them sub-properties of usedFor as defined by RDFS. Provided that no other extensions are required, the only thing they have to do is defining these three properties in RDFS. We introduce the prefix ex: for our colleague's thesaurus:

Sample Extension of Our Schema

```
ex:usedForAbbreviation rdf:type rdf:Property .
ex:usedForAbbreviation rdfs:subPropertyOf thes:usedFor .
```

[53] cf. DCMI, 2003

```
ex:usedForForeignTerm rdf:type rdf:Property .
ex:usedForForeignTerm rdfs:subPropertyOf thes:usedFor .

ex:usedForTrademark rdf:type rdf:Property .
ex:usedForTrademark rdfs:subPropertyOf thes:usedFor .
```

A computer application processing this RDF schema is redirected by means of URI to our schema and understands that usedForAbbreviation property defined by our colleague's schema IS-A usedFor property defined by our schema. Thanks to the 'everything is a resource' approach we can make use of all RDF properties and classes that are published on the Internet. You may consider this approach a kind of semantic counterpart to the hyperlink concept of the Web that allows us to create a link to every website that is accessable online.

RDF's data model is often compared with the data model of object-oriented (OO) programming languages. But note that what we have just done in RDF would have not been possible in an OO programming language since for the latter one the definition of properties (in OO languages referred to as attributes) is part of the class definition. If RDF had adopted this data model a property could have been only defined by modifying the definition of that class whose instances may use these properties. By definining classes and properties independently of each other, RDF allows us to define properties without having to change the class definition, i.e. to modify an existing schema. In contrast to OO programming languages the use of properties is not restricted by means of the appropriate class definitions but by means of the RDFS domain and range properties that reside on the property definitions.

Consider the last example again: The domain and range that we have defined for the super-property do apply to its sub-properties, i.e. usedForAbbreviation may only connect instances of the class thes:term. As a matter of course our colleagues are free to define additional properties that are not sub-properties of existing properties. They could for example define the property antonymOf in the following way:

Another Sample Extension of our Schema

```
ex:antonymOf  rdf:type  rdf:Property .
ex:antonymOf  rdf:domain  thes:term .
ex:antonymOf  rdf:range thes:term .
```

In this way thesaurus developers can easily define new classes and properties that are not defined yet. The more RDF definitions exist the more likely it is that the class or property one is searching for is already defined by someone else. Instead of writing a new grammar for each application as it would be required if we used XML-DTDs we combine already defined elements from an unlimited number of grammars.

The resource concept is not only useful when it comes to schema definitions but also when it comes to RDF descriptions. Suppose we want to express that between two terms of different thesauri exist a hierarchical relationship. No problem at all in RDF:

A Statement using Terms from three Vocabularies

```
ex1:eye thes:narrowerTerm ex2:head .
```

This example provides that both ex1:eye and ex2:head are instances of thes:term since thes:narrowerTerm specifies thes:term as domain and range. If we had not specified a domain and range thes:narrowerTerm could have connected any classes. That is why we should use the domain and range properties very carefully.

In fact, this thesaurus integration is not only a theoretical discussion but realized in several projects. The CERES RDF schema for thesauri that we have discussed above was developed in order to integrate the vocabularies of several thesauri in the field of environmental science. Probably the most ambitious attempt to integrate thesauri is done by the National Library of Medicine (NLM) of the United States. Since 1986 it develops a Unified Medical Language System (UMLS) [54] whose UMLS Metathesaurus integrates by now more than 100 biomedical vocabularies and classifications. It strikes that they do not use RDF for this purpose but an relational database management system and JAVA.

5.5 RDF and XML

As stated earlier, RDF does not come with its own syntax. The triple syntax we have used throughout this dissertation serves its purpose when it comes to theoretical discussions, but unfortunately it is not machine-readable. On principle it is up to every application programmer to choose the appropriate syntax for RDF, but the W3C strongly recommends to make use of XML. As a matter of fact XML seems to be meant for a syntax for RDF for it supports all required features like URI and namespaces [55], i.e. the use and distinction of terms from different vocabularies by means of prefixes. This match is not a coincidence since RDF is meant for the Web whose principles have to be supported as a matter of course by its main standard, i.e. XML. The XML syntax that is used in order to encode RDF representation is normally referred to as RDF/XML. It might strike the reader to get to know that there are several abbreviated syntaxes of RDF/XML available since different syntaxes produce different XML trees and we considered this the main reason for preferring RDF to XML for thesaurus representation. But it is crucial to understand that RDF/XML *represents* statements in RDF and *serializes* them in XML. An RDF application processing RDF/XML documents will therefore read the XML serialization and understand the RDF representation. That is why all RDF/XML syntaxes produce different XML trees but the same internal RDF models. It is beyond the scope of this dissertation to discuss RDF/XML and its different flavours in detail. Having understood the triple syntax it should be no problem to understand the RDF/XML syntax. As RDF/XML is the pick of all available syntaxes the interested reader will find the RDF schema for thesauri in RDF/XML syntax in appendix 4. A sample RDF description of a previous discussed ERIC descriptor entry can be found in appendix 5.

54) cf. UMLS, 2003

55) cf. Namespaces, 1999

PART 6
After All

We described the theory of terms, concepts and relationships of a thesaurus. We gave reasons for representing thesauri in RDF by explaining the shortcomings of native XML representations. We found out that RDF is an appropriate language for the representation of thesauri although we did not manage to find a proper RDF representation for the combination of single-word descriptors for compound terms. Nevertheless we consider the proposed RDF schema for thesauri superior to the competing schema of the CERES/NBII project since ours implements the preferable concept-based approach and corrects some of the design failures of CERES' schema. Besides our schema complies with the current RDF specification.

Having done this, the critical reader might confront us with the question if after all we facilitated the tremendous task of indexing a vast amount of books and websites. Unfortunately we have to deny this question. Our schema does not change the fact that indexing remains a time-consuming and unpleasent task that has to be done by human beings - regardless of the format of the documents that we have to index. So what is our RDF schema good for?

By showing that the abstract model of thesauri and the data model of RDF match, we have paved the way for the integration of various vocabularies that evolved independently of each other. Bearing the example of our introduction in mind, we could say that instead of forcing our librarian to adopt a more widely used and more extensive wordlist, which would imply the reindexing of all books, we could connect his vocabulary with the thesauri of other libraries. And by doing so, we would create a networked vocabulary that could be not only used for the integration of bibliographic data but also for indexing the Web.

We mentioned in the beginning that it would be very unlikely if all indexers of the world agreed on a single wordlist; we also explained why the the native XML representation of controlled vocabularies is not preferable for the same lack of agreement. Thanks to its consistent resource-based approach RDF allows us to pick the appropriate tools from various tool boxes instead of having to buy a prearranged one. That's why RDF schemas are more likely to succeed where others failed. Nevertheless our schema remains a standard that requires a certain level of acceptance in order to be useful. It is up to the thesaurus developers to test its applicability. Maybe this dissertation convinced them to try it.

PART 7

Bibliography

ANSI/NISO Z39.19, 1993:
Guidelines for the Construction, Format, and Management of Monolingual Thesauri
Developed by the National Information Standards Organization, approved by the American National Standards Institute

ANSI/NISO Z39.50, 1995:
Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. - Developed by the National Information Standards Organization, approved by the American National Standards Institute
Also available from http://www.loc.gov/z3950/agency/index.html [Last update: 2003-03-18]

APAIS 2002:
National Library of Australia: Thesaurus of Australian Public Affairs Information Service. - Version 2.0. - 2002-05
http://www.nla.gov.au/apais/thesaurus/index.html [Cited on: 2003-06-03]

Berners-Lee, 1998:
Why RDF model is different from the XML model. - 1998-10-14
http://www.w3.org/DesignIssues/RDF-XML.html

Burkart, 1997:
Burkart, Margarete: Thesaurus
In: Grundlagen der praktischen Information und Dokumentation / Marianne Bruder ... (Hrsg.). - völlig neu gefasste Ausgabe. - München [u.a.] :
Saur, 1997

CERES, 2003:
The California Environmental Resources Evaluation System (CERES) : Home Page. - http://ceres.ca.gov/index.html [Cited on: 2003-06-03]

CERES/NBII, 1998:
The CERES/NBII Thesaurus Partnership Project:
Thesaurus::RDF : The RDF Thesaurus descriptor standard. - [ca. 1998]
http://ceres.ca.gov/thesaurus/RDF.html

CERES/NBII, 2000:
The CERES/NBII Thesaurus Partnership Project : Home Page
http://ceres.ca.gov/thesaurus/index.html [Last update: 2000-09-26]

DAML+OIL, 2001:
World Wide Web Consortium (W3C):
DAML+OIL Reference Description ; W3C Note / Dan Connolly ;
Frank van Harmelen ; Ian Horrocks ; Deborah L. McGuinness ; Peter F.
Patel-Schneider ; Lynn Andrea Stein. - 2001-12-18
http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218

DCMI, 2003:
Dublin Core Metadata Initiative : Home Page
http://dublincore.org/index.shtml [Cited on: 2003-06-03]

DIN, 1463 1987-11:
Erstellung und Weiterentwicklung von Thesauri: Einsprachige Thesauri. -
Norm des Deutschen Instituts für Normung (DIN)

ERIC, 2003:
Educational Resources Information Center (ERIC):
Thesaurus of ERIC Descriptors
http://www.ericfacility.net/extra/pub/thessearch.cfm [Cited on: 2003-06-03]

IEST, 1998:
International Energy Agency: International Energy Subject Thesaurus (IEST). -
2nd Revision. - 1998
http://www.etde.org/edb/download.html [Last update: 2003-05]

ISO 2788, 1986 2nd edition:
Guidelines for the establishment and development of monolingual thesauri. -
Standard of the International Organization for Standardization (ISO)

Maniez, 1988:
Maniez J.: Relationships in thesauri : Some critical remarks
In: International Classification, 15 (1988), pp. 133-138

MeSH, 2003:
National Library of Medicine: Medical Subject Headings (MeSH). - Version 2003
http://www.nlm.nih.gov/mesh/meshhome.html [Last update: 2002-03-05]

Namespaces, 1999:
World Wide Web Consortium (W3C):
Namespaces in XML ; W3C Recommendation / Editors: Tim Bray (Textuality) ;
Dave Hollander (Hewlett-Packard Company) ; Andrew Layman (Microsoft). -
1999-01-14
http://www.w3.org/TR/1999/REC-xml-names-19990114

Nelson et al., 2001:
Nelson, Stuart J. ; Johnston, Douglas W. ; Humphreys, Betsy L.:
Relationships in Medical Subject Headings
In: Bean, Carol A. ; Green ; Rebecca (eds):
Relationships in the organization of knowledge. - New York : Kluwer
Academic Publishers, 2001. - pp. 171-184
Also available from: http://www.nlm.nih.gov/mesh/meshrels.html [Last update: 2001-11-20]

RDF, 1999:
World Wide Web Consortium (W3C):
Resource Description Framework (RDF) : Model and Syntax Specification ; W3C Recommendation /
Editors: Ora Lassila (Nokia Research Center) ; Ralph R. Swick (W3C). - 1999-02-22
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ [Last update: 1999-02-24]

RDF Primer, 2003:
World Wide Web Consortium (W3C):
RDF Primer ; W3C Working Draft / Editors: Frank Manola (The Mitre Corporation) ; Eric Miller (W3C). - Version 2003-01-23
http://www.w3.org/TR/2003/WD-rdf-primer-20030123/

RDFS, 2003:
World Wide Web Consortium (W3C):
RDF Vocabulary Description Language 1.0: RDF Schema ; W3C Working Draft /
Editors: Dan Brickley (W3C/ILRT) ; R.V. Guha (IBM). - 2003-01-23
http://www.w3.org/TR/2003/WD-rdf-schema-20030123/

RFC 2141, 1997-05:
URN Syntax. - Request for Comments (RFC) of the Internet Engineering Task Force
http://www.ietf.org/rfc/rfc2141.txt

RFC 2396, 1998-08:
Uniform Resource Identifiers (URI): Generic Syntax. -
Request for Comments (RFC) of the Internet Engineering Task Force
http://www.ietf.org/rfc/rfc2396.txt

TEST, 1967:
US Department of Defense:
Thesaurus of Engineering and Scientific Terms. - 1967

UMLS, 2003:
National Library of Medecine:
Unified Medical Language System (UMLS) : Home Page
http://umlsinfo.nlm.nih.gov/ [Cited on 2003-06-03]

W3C, 2003:
World Wide Web Consortium (W3C) : Home Page
http://www.w3c.org/ [Last update 2003-06-02]

XML, 2000:
World Wide Web Consortium (W3C):
Extensible Markup Language (XML) 1.0 (Second Edition) ; W3C Recommendation / Editors: Tim Bray (Textuality and Netscape) ; Jean Paoli (Microsoft) ;
C.M. Sperberg-McQueen (University of Illinois at Chicago and Text Encoding Initiative) ; Eve Maler (Sun Microsystems, Inc.). - 2000-10-06
http://www.w3.org/TR/2000/REC-xml-20001006

XPath, 1999:
World Wide Web Consortium (W3C):
XML Path Language (XPath): Version 1.0 ; W3C Recommendation. - 1999-11-16
http://www.w3.org/TR/1999/REC-xpath-19991116

XSchema, 2001:
World Wide Web Consortium (W3C):
XML Schema ; W3C Recommendation. - 2001-05-02
http://www.w3.org/XML/Schema [Last update: 2003-01-01]

Zthes, 2001:
Zthes : a Z39.50 Profile for Thesaurus Navigation. - Version 0.5. - 2001-11-06
http://zthes.z3950.org/profile/zthes-05.html [Last update 2002-11-28]

PART 8

Appendix

8.1 Appendix: DTD of the Zthes Markup Language

Copied from Zthes, 2001

```
<!-- Zthes DTD, version 0.5
     Based on Z39.50 Profile for Thesaurus Navigation, version 0.5
     Incorporating modifications by National Library of Australia
     Date of DTD: 19 Oct 2001 -->

<!-- #PCDATA: parseable character data = text

     occurence indicators (default: required, not repeatable):
     ?: zero or one occurrence (optional)
     *: zero or more occurrences (optional, repeatable)
     +: one or more occurrences (required, repeatable)

     |: choice, one or the other, but not both
 -->

<!ENTITY % terment         "termId, termName, termQualifier?,
                            termType?, termLanguage?">

<!ENTITY % admin           "termCreatedDate?, termCreatedBy?,
                            termModifiedDate?, termModifiedBy?">

<!ELEMENT Zthes            (term+)>

<!ELEMENT term             (%terment;, termNote?,
                            %admin;,
                            relation*)>

<!ELEMENT relation         (relationType, sourceDb?, %terment;)>

<!ELEMENT termId           (#PCDATA)>
<!ELEMENT termName         (#PCDATA)>
<!ELEMENT termQualifier    (#PCDATA)>
<!ELEMENT termType         (#PCDATA)>
<!ELEMENT termLanguage     (#PCDATA)>
<!ELEMENT termNote         (#PCDATA)>
<!ELEMENT termCreatedDate  (#PCDATA)>
<!ELEMENT termCreatedBy    (#PCDATA)>
<!ELEMENT termModifiedDate (#PCDATA)>
<!ELEMENT termModifiedBy   (#PCDATA)>
<!ELEMENT relationType     (#PCDATA)>
<!ELEMENT sourceDb         (#PCDATA)>
```

8.2 Appendix: RDF Schema for Thesauri in Triple Notation

```
(Definition of Classes)


thes:ThesaurusItem  rdf:type  rdf:Class .

thes:Concept  rdf:type rdf:Class .
thes:Concept  rdf:subClassOf thes:ThesaurusItem .

thes:Term rdf:type  rdf:Class .
thes:Term rdf:subClassOf  thes:ThesaurusItem .


(Properties of ThesaurusItem)

thes:source rdf:type  rdf:Property .
thes:source rdfs:domain thes:ThesaurusItem .
thes:source rdfs:range rdfs:Literal .

thes:historyNote rdf:type rdf:Property .
thes:historyNote rdfs:domain thes:ThesaurusItem .
thes:historyNote rdfs:range rdfs:Literal .

thes:dateOfAddition rdf:type rdf:Property .
thes:dateOfAddition rdfs:domain thes:ThesaurusItem .
thes:dateOfAddition rdfs:range rdfs:Literal .

thes:dateOfDeletion rdf:type rdf:Property .
thes:dateOfDeletion rdfs:domain thes:ThesaurusItem .
thes:dateOfDeletion rdfs:range rdfs:Literal .


(Properties of Term)

thes:label rdf:type rdf:Property .
thes:label rdfs:domain  thes:Term .
thes:label rdfs:range rdfs:Literal

thes:type rdf:type rdf:Property .
thes:type rdfs:domain  thes:Term .
thes:type rdfs:range rdfs:Literal .

thes:scopeNote rdf:type rdf:Property .
thes:scopeNote rdfs:domain  thes:Term .
thes:scopeNote rdfs:range rdfs:Literal .

thes:category rdf:type rdf:Property .
thes:category rdfs:domain  thes:Term .
thes:category rdfs:range rdfs:Literal .

thes:definition rdf:type rdf:Property .
thes:definition rdfs:domain  thes:Term .
thes:definition rdfs:range rdfs:Literal .


(Relationship between Term and Concept)

thes:belongsToConcept rdf:type rdf:Property .
thes:belongsToConcept rdfs:domain thes:Term .
thes:belongsToConcept rdfs:range thes:Concept .
```

```
thes:comprisesTerm rdf:type rdf:Property .
thes:comprisesTerm rdfs:domain thes:Concept .
thes:comprisesTerm rdfs:range thes:Term .


(Associative Relationship)

thes:relatedConcept rdf:type rdf:Property .
thes:relatedConcept rdfs:domain thes:Concept .
thes:relatedConcept rdfs:range thes:Concept .


(Equivalence Relationship)

thes:usedFor rdf:type rdf:Property .
thes:usedFor rdfs:domain thes:Term .
thes:usedFor rdfs:range thes:Term .

thes:use rdf:type rdf:Property .
thes:use rdfs:domain thes:Term .
thes:use rdfs:range thes:Term .


(Hierarchical Relationship)

thes:broaderConcept rdf:type rdf:Property .
thes:broaderConcept rdfs:domain thes:Concept .
thes:broaderConcept rdfs:range thes:Concept .

thes:narrowerConcept rdf:type rdf:Property .
thes:narrowerConcept rdfs:domain thes:Concept .
thes:narrowerConcept rdfs:range thes:Concept .

thes:broaderConceptGeneric rdf:type rdf:Property .
thes:broaderConceptGeneric rdfs:domain thes:Concept .
thes:broaderConceptGeneric rdf:range thes:Concept .

thes:narrowerConceptGeneric rdf:type rdf:Property .
thes:narrowerConceptGeneric rdfs:domain thes:Concept .
thes:narrowerConceptGeneric rdf:range thes:Concept .

thes:broaderConceptPartitive rdf:type rdf:Property .
thes:broaderConceptPartitive rdfs:domain thes:Concept .
thes:broaderConceptPartitive rdf:range thes:Concept .

thes:broaderConceptInstance rdf:type rdf:Property .
thes:broaderConceptInstance rdfs:domain thes:Concept .
thes:broaderConceptInstance rdf:range thes:Concept .
```

8.3 Appendix: CERES' RDF Schema for Thesauri

copied from CERES/NBII, 1998

```
<rdf:RDF
     xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#";
     xmlns:rdfs="http://www.w3.org/TR/WD-rdf-schema#";>

  <rdfs:Class ID="Term">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Resource";/>
  </rdfs:Class>

  <rdf:PropertyType ID="HN">
    <rdf:domain rdf:resource="#Term"/>
    <rdf:range
rdf:Resource="http://www.w3.org/TR/WD-rdf-syntax#String";/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="Source">
    <rdf:domain rdf:resource="#Term"/>
    <rdf:range
rdf:Resource="http://www.w3.org/TR/WD-rdf-syntax#String";/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="Status">
    <rdf:domain rdf:resource="#Term"/>
    <rdf:range
rdf:Resource="http://www.w3.org/TR/WD-rdf-syntax#String";/>
  </rdf:PropertyType>

  <rdfs:Class ID="Category">
    <rdfs:subClassOf rdf:resource="Term"/>
  </rdfs:Class>

  <rdf:PropertyType ID="Descriptor">
    <rdf:domain rdf:resource="#Category"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdfs:Class ID="Descriptor">
    <rdfs:subClassOf rdf:resource="Term"/>
  </rdfs:Class>

  <rdf:PropertyType ID="SN">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range
rdf:Resource="http://www.w3.org/TR/WD-rdf-syntax#String";/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="CN">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range
rdf:Resource="http://www.w3.org/TR/WD-rdf-syntax#String";/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="CAT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Category"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="TT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="BT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="RT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="NT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="LT">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  <rdf:PropertyType ID="UF">
    <rdf:domain rdf:resource="#Descriptor"/>
    <rdf:range rdf:Resource="#EntryTerm"/>
  </rdf:PropertyType>

  <rdfs:Class ID="EntryTerm">
    <rdfs:subClassOf rdf:resource="Term"/>
  </rdfs:Class>

  <rdf:PropertyType ID="USE">
    <rdf:domain rdf:resource="#EntryTerm"/>
    <rdf:range rdf:Resource="#Descriptor"/>
  </rdf:PropertyType>

  </rdf:RDF>
```

8.4 Appendix: RDF Schema for Thesauri in RDF/XML

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

<!-- Class Definitions -->

<rdf:Description rdf:ID="ThesaurusItem">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
</rdf:Description>

<rdf:Description rdf:ID="Term">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
    <rdfs:subClassOf rdf:resource="#ThesaurusItem">
</rdf:Description>

<rdf:Description rdf:ID="Concept">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
    <rdfs:subClassOf rdf:resource="#ThesaurusItem">
</rdf:Description>

<!-- Properties of ThesaurusItem -->

<rdf:Description rdf:ID="source">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#ThesaurusItem"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="historyNote">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#ThesaurusItem"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="dateOfAddition">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#ThesaurusItem"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="dateOfDeletion">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#ThesaurusItem"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<!-- Properties of Term -->
```

```
<rdf:Description rdf:ID="label">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<!-- entryTerm or preferredTerm or categoriser (non-MeSH) -->

<rdf:Description rdf:ID="type">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="scopeNote">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="category">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<rdf:Description rdf:ID="definition">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range
rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Description>

<!-- Relationship between Term and Concept -->

<rdf:Description rdf:ID="belongsToConcept">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="comprisesTerm">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Term"/>
</rdf:Description>

<!-- Associative Relationship -->

<rdf:Description rdf:ID="relatedConcept">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>
```

```
<!-- Equivalence Relationship -->

<rdf:Description rdf:ID="usedFor">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range rdf:resource="#Term"/>
</rdf:Description>

<rdf:Description rdf:ID="use">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range rdf:resource="#Term"/>
</rdf:Description>

<!-- Hierachical Relationship -->

<rdf:Description rdf:ID="broaderConcept">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="narrowerConcept">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="broaderConceptGeneric">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#broaderConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="narrowerConceptGeneric">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#narrowerConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="broaderConceptPartitive">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#broaderConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="narrowerConceptPartitive">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#narrowerConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="broaderConceptInstance">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#broaderConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>

<rdf:Description rdf:ID="narrowerConceptInstance">
    <rdf:type
rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:subPropertyOf rdf:resource="#narrowerConcept">
    <rdfs:domain rdf:resource="#Concept"/>
    <rdfs:range rdf:resource="#Concept"/>
</rdf:Description>
```

8.5 Appendix: Description of Sample Descriptor Entry

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:thes="http://www.agnosis.de/rdfs/thesaurus#">

  <rdf:Description rdf:ID="C1">
    <rdf:type
rdf:resource="http://www.agnosis.de/rdfs/thesaurus#Concept"/>
    <thes:comprisesTerm thes:label="Activism"
      thes:type="preferredTerm" rdf:resource="T1"/>
    <thes:comprisesTerm thes:label="Militancy"
      rdf:resource="#T2"/>
    <thes:comprisesTerm thes:label="Political Protest"
      rdf:resource="#T3"/>
    <thes:broaderConcept thes:label="Social Behaviour"
      rdf:resource="#C2"/>
    <thes:relatedConcept thes:label="Alienation"
      rdf:resource="#C3"/>
    <thes:relatedConcept thes:label="Citizen Participation"
      rdf:resource="#C4"/>
    <thes:relatedConcept thes:label="Civil Disobedience"
      rdf:resource="#C5"/>
    <thes:relatedConcept thes:label="Demonstrations (Civil)"
      rdf:resource="#C6"/>
    <thes:relatedConcept thes:label="Dissent"
      rdf:resource="#C7"/>
    <thes:relatedConcept thes:label="Lobbying"
      rdf:resource="#C8"/>

  <rdf:Description rdf:ID="T1">
    <rdf:type
rdf:resource="http://www.agnosis.de/rdfs/thesaurus#Term"/>
    <thes:label>Activism</thes:label>
    <thes:type>preferredTerm</thes:type>
    <thes:belongsToConcept rdf:resource="#C1"/>
    <thes:scopeNote>Movements and procedures designed to force
      changes in rules and practices or to hasten social
      change</thes:scopeNote>
    <thes:usedFor thes:label="Militancy"
      rdf:resource="#T2"/>
    <thes:usedFor thes:label="Political Protest"
      rdf:resource="#T3"/>
    <thes:dateOfAddition
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    1969-01</thes:dateOfAddition>
  </Description>

  <rdf:Description rdf:ID="T2">
    <rdf:type
rdf:resource="http://www.agnosis.de/rdfs/thesaurus#Term"/>
    <thes:label>Militancy</thes:label>
    <thes:type>entryTerm</thes:type>
    <thes:belongsToConcept rdf:resource="#C1"/>
    <thes:use thes:label="Activism" rdf:resource="#T1"/>
  </Description>

  <rdf:Description rdf:ID="T3">
    <rdf:type
rdf:resource="http://www.agnosis.de/rdfs/thesaurus#Term"/>
    <thes:label>Political Protest</thes:label>
    <thes:type>entryTerm</thes:type>
    <thes:belongsToConcept rdf:resource="#C1"/>
    <thes:use thes:label="Activism" rdf:resource="#T1"/>
  </Description>

</RDF>
```

Colophon

This dissertation was written in Simplified DocBook (DTD V4.1.2.5) and converted to PDF using the Formatting Objects Processor (V0.20.4) of the Apache XML Project and an XSL stylesheet written by the author. The text was edited with Marko Macek's FTE editor (V0.46). For the layout of this dissertation various books published by O'Reilly served as an example.