2 Knowledge Management

This chapter deals with “Knowledge”, “Knowledge Management” and “Knowledge Management technologies”.

Key concepts for Knowledge Management as semantic networks and ontologies are introduced, followed by relevant standards in that area. An overview of search- and retrieval-methods is given in this Chapter.

Section 2.3 introduces the term “Knowledge Management System” and shows the technologies and applications forming a Knowledge Management System followed by a review of four Knowledge Management Systems. These KM systems have been selected to illustrate the spectrum rather than a full overview of the market.

2.1 What is Knowledge?

Although everyone is familiar with the term “knowledge”, there is no commonly accepted definition [Bodrow & Bergmann, 2003]. Definitions depend on the subjective point of view of the person defining the term. In many common definitions [Aamodt & Nygard, 1995] [Davenport & Prusak, 2000] [Kornwachs, 2001] [Nonaka & Takeuchi, 1995] [North, 2005] [Probst et al., 2000] we find three types of statements:

- Knowledge is based on information.
- All information is related to each other in coherence.
- Knowledge must be in coherence with the perception environment.

The knowledge pyramid (see Figure 2.1-1) [Wolf et al., 1999] [Aamodt & Nygard, 1995] shows that human comprehension is based on the extraction of information from data and on the allocation of information in a certain context.

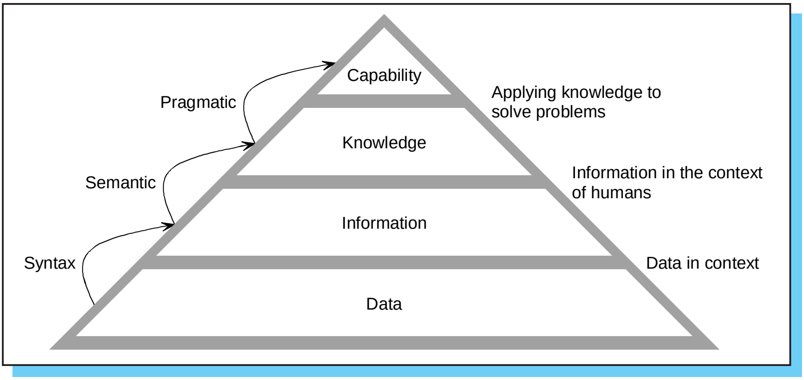

Figure 2.1-1 Knowledge Pyramid

There are many philosophical theories of knowledge that discuss the limits of what we can know [Gardner, 1985] [Musgrave, 1993] [Rich, 1981] [Scruton, 1995] [Spender, 1996]. This research concentrates more on the technological view of managing knowledge.

Data contains characters and figures without any context. They become information by adding a context. Knowledge results from information processing by humans and is "recognised information". Humans are using this knowledge to control their activities. [North, 2002]

Knowledge results of building networks of information and allows a human being to generate the capability to interact with his environment [Haun, 2002].

Probst et al. define knowledge as the entirety of all cognitions and capabilities that a human being uses to solve problems [Probst et al., 1999].

North shows the process from data to competence and in addition to **competitiveness** of a company. He states that knowledge is only of value if it is transformed to ability. This can only be realised by the active processing of knowledge by a person. Generating knowledge using information is named as "**competence**" by North, 2002.

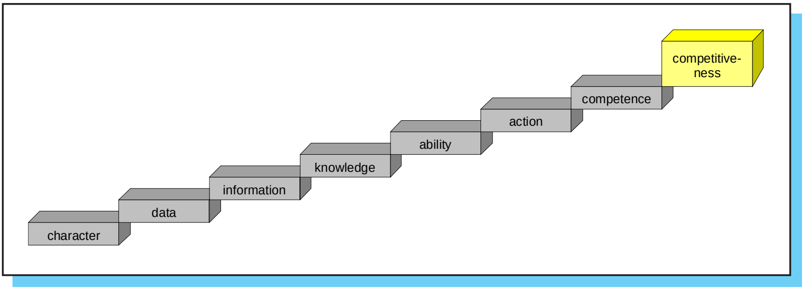

Figure 2.1-2 Knowledge Staircase [North, 2002]

Knowledge can be classified in several ways. In this thesis I use the following classification according to [Haun, 2002]:

- **Procedural knowledge** captures procedural methods and strategies. It is also named as “know-how”.
- **Expert knowledge** is built by sensory perception in a special situation.
- **Declarative knowledge** represents awareness of reality. It describes facts, regularities or circumstances and is also named as “know-that” or “know-what”.
- **Statistical knowledge**, **Causal knowledge** or **Processing knowledge** represents collections of circumstances. Causal knowledge memorises motivation and reason, also named as “know-why”.
- **Heuristic knowledge** defines rules for certain circumstances.
- **Classification knowledge** enables to classify complex circumstances.
- **Relational knowledge** allows seeing structures and correlations of complex circumstances.

Depending on the availability of the knowledge for one or more people, knowledge can be differentiated into:

- **Individual knowledge** is bound to the owner and can be called “private”.
- **Collective knowledge** describes knowledge of a number of persons. If it is available not only for a group but also for a whole company it is called **Organisational knowledge**.

In general, there are two kinds of knowledge [Nonaka & Takeuchi, 1995]:

- **Explicit knowledge** can be described and formalised. It can be structured, phrased and written down in documents. These facts can be stored in documents, databases and other IT systems.
- **Tacit knowledge** is bound to people. It is subjective and normally it can't be formalised completely in an easy way. Also it relates to special contexts and can be communicated poorly. There is a lot of important tacit knowledge within companies including expert knowledge, know-how, ability, capability and competence. As an example for tacit knowledge one can try to describe the sound of a guitar. All facts can be known like the frequencies of all notes (e.g. A = 440 Hz). However, if you haven't heard a guitar playing you can't imagine the sound.

Figure 2.1-3 shows an overview of types of knowledge.

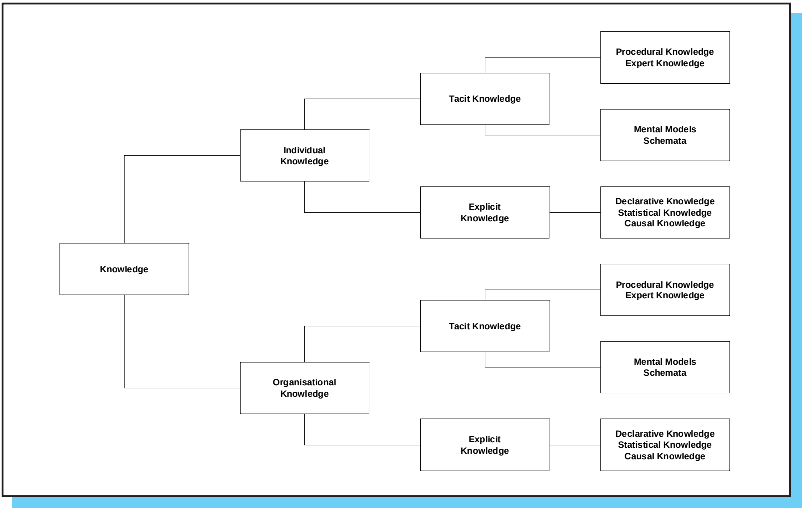

Figure 2.1-3 Types of Knowledge [Haun, 2002]

Further details concerning the definition of knowledge will be given in Section 3.4 and 3.5.

The definition of knowledge used in this thesis

In my understanding, knowledge is information in the context of a specific person. Probst et al. also state that knowledge is based on data and information but always bound to people [Probst et al., 1999].

2.2 Knowledge Management Concepts

The technological oriented realisation of Knowledge Management bases on a few key concepts for knowledge structures: ontologies and their technological representations, semantic networks for example. Section 2.2.2 gives an introduction into these concepts.

Section 2.2.3 provides an overview of relevant standards in the Knowledge Management area. These standards are used for the formal representation of knowledge structures. I distinguish here the family of "Semantic Web" standards and the "Topic Map" standard.

For searching and navigation within knowledge structures an overview of search- and retrieval-methods is given in Section 2.2.4.

The basis for managing knowledge is the structuring of contents. Within companies there have been many approaches suggested to organize knowledge. These approaches can be divided into two groups [Maier, 2002]:

- ontology-based, defined in Section 2.2.2 below,
- based on business process models.

2.2.1 KM based on Business Process Models

The approach based on business process models uses these representations to identify most critical business knowledge [Remus, 2001]. Though this can be of value for companies who have well defined models for all relevant business processes, it eliminates a lot of highly creative processes in research and development or creating new company strategies. Therefore, I propose to use existing business process models or existing workflow descriptions to identify concepts of importance and manage them using an ontology based approach. This increases the flexibility in managing knowledge in comparison to systems based on workflow management systems that execute the business processes and manage additional knowledge. In addition all knowledge that is not connected with business processes can be managed within an ontology-based approach (e.g., knowledge concerning future products). In the developed ontology-based knowledge pool the relations of concepts and business processes and workflows should be documented.

2.2.2 Semantic Networks and Ontologies

An **ontology** is a formal, explicit specification of a commonly used abstract model of the world [Gruber, 1995]. It allows the representation of knowledge structures [Maier, 2002].

Different definitions of the term ontolgy exist, mostly with a philosophical background.

In this thesis, I use the following definition [Gruber, 1995 p908f]:

> *"An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence.*
>
> ...
>
> *When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge.*
>
> ...
>
> *Formally, an ontology is the statement of a logical theory."*

An updated and detailed definition of the term ontology can be found in Gruber, 2008.

The relation to the application domain has been introduced by Sowa, 2000: "Ontology defines the kinds of things that are existing in the application domain."

Ontologies can be technically implemented in different ways. Standards for ontologies are Resource Description Framework Schema (RDF-Schema), Web Ontology Language (OWL) and Topic Maps. They will be specified in Section 2.2.3. A special ontological structure is a semantic network. This concept forms the basis of the system presented in this thesis.

Semantic networks are directed graphs, consisting of nodes, node labels, links and link labels [Helbig, 2005]. A node describes an object of the real world. Links are representing relationships between two nodes. A link label describes the type of the relationship and the direction of that link [Winston, 1993]. The idea of semantic networks goes back to M. Ross Quillian [Quillian, 1968] to model human memory in relation to the area of cognitive psychology. Semantic networks should enable graphical representation of knowledge that is existing in natural language.

As shown in Figure 2.2-1, semantic networks allow the graphical visualisation of information. All imaginable ontologies can be represented by semantic networks. Most common kinds of semantic networks are [Sowa, 2006]:

- Definitional networks (described below);
- Assertion networks (designed to assert propositions, proposed as models of the conceptual structures underlying natural language semantics);
- Implicational networks (implication as the primary relation for connecting nodes);
- Executable networks (include mechanism for performing inferences, passing messages, or searching for patterns and associations);
- Learning networks (build or extend their representations by acquiring knowledge from examples);
- Hybrid networks (combinations of above mentioned techniques).

In order to represent internal company information, I recommend the so-called “Definitional Networks” – a special form of semantic networks. “Definitional networks emphasize the subtype or is-a relation between a concept type and a newly defined subtype. The resulting network, also called a generalization or subsumption hierarchy, supports the rule of inheritance for copying properties defined for a supertype to all of its subtypes.” [Sowa, 2006] They are suitable for the construction of ontologies, especially for class and generalization hierarchies.

Frequently used relation types are:

- IS-A / AKO (a kind of), for representation of inheritance and generalization hierarchies,
- PART-OF / HAS-A-PART for representation of aggregation,
- MEMBER-OF / INSTANCE-OF for individualization.

A semantic network consists of

- Nodes: representation of concepts that are semantic entities (objects or attributes) of the real world that are explicit and unique,
- Node label: name of concept,
- Links: semantic relations between concepts. Assigning attributes to objects. Indicates direction of links,
- Link label: describes the relationship of that link.

Semantic networks allow generic links like “connected-to”. It depends on the implementation of the used system whether it offers built-in generic links or the user has to define them.

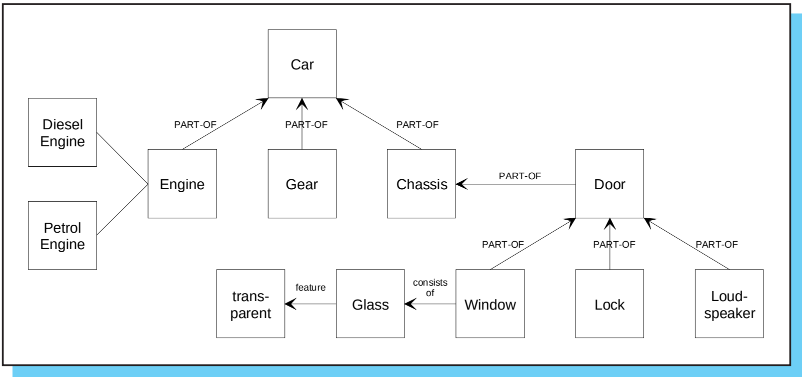

Figure 2.2-1 Example of a Definitional Network (part)

Reification
Most kinds of semantic networks allow quoting so that statements about statements can be made. This is called Reification [Newcomb, 2002]. An example of reification in RDF is shown in Figure 2.2-3.

Advantages / Disadvantages of Semantic Networks
The main advantage of semantic networks is the possibility for a graphical representation. Most semantic network - based tools for the design of knowledge structures are using some kinds of graphical representation within their user interface.

On the other hand defining a semantic network in such a graphical way prevents a unique formal representation in one of the standards decribed in the next section. For example not all RDF Graphs can be expressed in RDF/XML [Becket, 2004].

The formal requirements of semantic networks raises the effort of adding new nodes to the network structure. Every node needs a label and a link to at least one existing node with a link direction and a link label. Knowledge structures like Mind Maps [Buzan & Buzan, 2003] or Topic Maps (described below) allow to add nodes without link direction and link labels (or even without a node label) which saves time especially within creative processes. Some tools for defining semantic networks allow the user also to drop link direction and link label (as illustrated in Figure 2.2-1 by the examples of "Diesel Engine" and "Petrol Engine") by using internal generic links for a standard-conform representation.

2.2.3 Standards

When using a Knowledge Management System *within* a company, it would seem that using standards is not mandatory. However, most Knowledge Management Systems offer interfaces to connect with other systems and to save and backup the stored data. For exchanging data between different Knowledge Management Systems via export / import interfaces, most systems use a standardised format like RDF. The next step is not only exchanging information but sharing and collaborating based on this information.

“The distribution of applications, however, has more complex needs. You need agreed protocols and interfaces between distributed software components and, last but not least, the data exchanged by these components must be machine-readable and understandable.” [Fensel et al., 2007]

The most interesting trend in this area is the “Semantic Web” that will be discussed below.

Many standards in the area of Knowledge Management are based on XML briefy described in Appendix B.

Semantic Web

“The Semantic Web provides a common framework that allows data to be shared and re-used across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF).” [W3C SW, 2007].

The vision behind the Semantic Web is to define and link data in ways that go beyond mere display in a Web browser, such as using it in machines for automated processes, integration and re-use. The processes involved should exchange and process data, although the programs have been designed and implemented independently. This vision could only be realised via standardisation.

Several groups are working on the following standardisation [W3C SW, 2007]:

- RDF standard;
- Gleaning Resource Descriptions from Dialects of Languages (GRDDL);
- SPARQL Query Language for RDF;
- Web Ontology Language (OWL);
- Semantic Web Services.

Resource Description Framework (RDF)

The W3C standard Resource Description Framework is designed to exchange meta data and to represent information about resources that can be identified within the web.

Concepts and abstract syntax of RDF are introduced in the W3C Recommendation by Klyne & Carroll, 2004.

Basic concept of RDF is so-called triples of subjects, predicates and objects (see Figure 2.2-2). A set of triples is called RDF-Graph.

An RDF-Graph is a directed graph consisting of

- subjects and objects as nodes,
- predicates (also called properties) as directed named links (also called arcs).

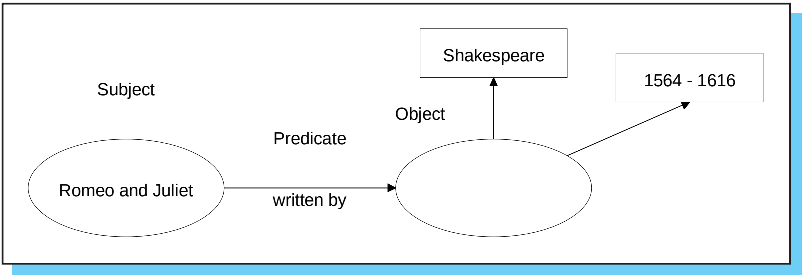

Figure 2.2-2 Example of RDF Graph

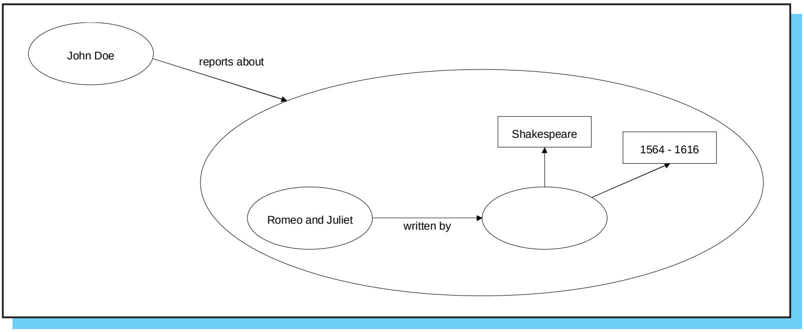

Figure 2.2-3 Example of reification in a RDF Graph

RDF uses reification as shown in Figure 2.2-3.

Nodes and links are represented by resources identified by an URI. An URI (Uniform Resource Identifier) is a unified concept for addressing of any kind of object in the Internet [Berners-Lee et al., 1998].

Subject and object represent either an URI reference or a string. Such a string is called Literal and will be shown as a rectangle. Resources are shown as ovals. A predicate can be an URI reference. It is called relation or property if the object represents an URI reference or an attribute if it is a literal. A URI reference refers to an RDF namespace that describe the properties meaning in detail. A RDF namespace is a set of RDF resources and their meanings – it provides a vocabulary. A namespace can refer to other namespaces.

There are two given namespaces in RDF:

- RDF Schema for the RDF vocabulary defined in the RDF namespace [RDF Vocabulary, 2007] that defines the basic constructs of RDF,
- RDF Schema vocabulary (RDFS) [RDF Schema Vocabulary, 2007] is a namespace for the creation of own vocabularies.

Since the RDF model is independent of the notation, there are two syntaxes: RDF/XML [Becket, 2004] and Notation 3 (N3) [Berners-Lee, 2006]. For exchanging data the RDF/XML format is recommended due to a higher level of awareness and market penetration of XML based formats.

N3 focuses on a compact and readable syntax to express data and logic in the same language. Rules can be expressed in N3 and reification can be modelled using a quote mechanism. [Berners-Lee, 2006]

RDF/XML is based on XML. It uses XML namespace [Tim Bray et al., 2006] so that XML namespace-qualified names (QNames) must be used which means restrictions on the set of legal characters. In practise not all RDF Graphs can be serialized in RDF/

XML, for example if used property names cannot be converted to XML namespace-qualified names or reserved names are used as property names [Becket, 2004].

```
<rdf:RDF>
xmlns:rdf=http://www.w3.org/TR/PR-RDF-Syntax#
xmlns:s="http://description.org/schema/">
    <rdf:Description about="Romeo and Juliet">
        <s:Author>
            <rdf:Description about="Author">
                <rdf:type resource=
                "http://example.org/schema/Person"/>
                <v:Name>Shakespeare</v:Name>
                <v:Lifetime>1564 - 1616</v:Lifetime>
            </rdf:Description>
        </s:Author>
    </rdf:Description>
</rdf:RDF>
```

Figure 2.2-4 Example of RDF /XML

Figure 2.2-4 shows the serialised RDF Graph of Figure 2.2-2.

The XML element “rdf:RDF” indicates that this element contains a statement.

“rdf:Description” describes the resource which is specified by the attribute “about”. The resource “Author” of Figure 2.2-14 is specified by two literals for the attributes “Name” and “Lifetime” using the “<v:ValueName>” syntax (v = value) as used in Lassila & Swick, 1999.

“rdf:type” assigns the author of Figure 2.2-4 to the type “Person”. The description of the class “Person” is available via the URL “http://example.org/schema/Person”.

RDF Schema (RDFS)

“RDF Schema ... is a collection of RDF resources that can be used to describe properties of other RDF resources (including properties) which define application-specific RDF vocabularies.” [Brickley & Guha, 1999] Resources are instances of one or more classes. In RDFS sub-classes are used to model hierarchies. The concepts of class, sub-class, and resource are shown in Figure 2.2-15.

Base classes in RDFS are “rdfs:Class”, “rdfs:Resource” and “rdfs:Property”.

The RDFS concept is comparable to object oriented programming (OOP) but do not focus on classes but on attributes. In OOP, classes are described using attributes. In RDFS, attributes are defined by classes. This simplifies statements about existing resources by assigning new attributes to classes using “rdfs:label”, “rdfs:comment” and “rdfs:seeAlso”. The class hierarchy of RDFS is shown in Figure 2.2-6, which is not complete and complies to an older version of RDFS but used here to illustrate the coherencies of the RDFS elements.

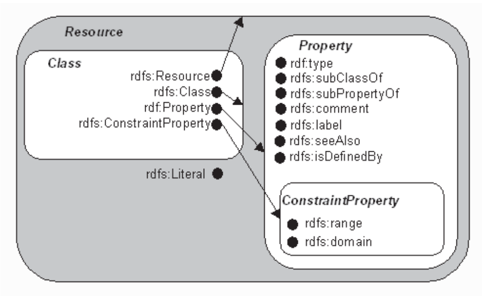

Figure 2.2-5 Classes and Resources as Sets and Elements [Brickley & Guha, 1999]

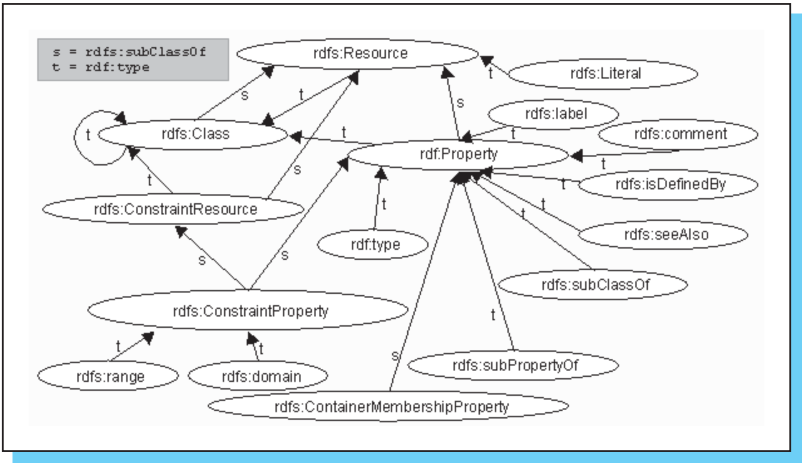

Figure 2.2-6 Class Hierarchy for the RDF Schema [Brickley & Guha, 1999]

For the complete RDFS recommendation of the W3C see Brickley & Guha, 2004.

RDF allows using linked lists where resources are elements of the list. There is no given structure for lists. More than one value can be assigned to an element of a list.

For grouping resources RDF offers three kinds of containers:

- Bag: unsorted set of resources; duplicates are possible;
- Alternative: choice 1 of n; the first element is standard value;
- Sequence: sorted set of resources.

Query Languages for RDF

There are some existing query languages for RDF but they are not standardised. RDQL - A Query Language for RDF [Seaborne, 2004] and SPARQL Protocol and RDF Query Language [Prud'hommeaux & Seaborne, 2008] are the most noted ones. In January 2008 SPARQL received the status of W3C Recommendation – a big step forward to a standard query language for RDF.

Web Ontology Language (OWL)

OWL is an ontology based language for the web based on the concepts of RDF and RDFS and adds more vocabulary for describing properties and classes. This includes:

- Relations between classes (e.g. disjointness: A,B are disjointed if the intersection of A and B is not empty);
- cardinality (e.g., ‘exactly one’);
- equality;
- richer typing of properties;
- characteristics of properties (e.g., symmetry);
- enumerated classes.

[W3C OWL, 2007]

OWL provides three sub-languages for use by specific implementers and users [McGuinness & van Harmelen, 2004]:

- **OWL Lite**: classification hierarchy and simple constraints only;
- **OWL DL** (DL => description logics): “maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class).” [McGuinness & van Harmelen, 2004]
- **OWL Full**: maximum expressiveness and the syntactic freedom of RDF with no computational guarantees; building an ontology by extending the pre-defined RDF or OWL vocabulary.

Ontology developers should choose the sublanguage that best suits their needs. If the more-expressive constructs for extending the existing vocabulary provided by OWL DL are not required OWL Lite should be used. OWL Full should be used only if the meta-modeling facilities of RDF Schema are required (e.g. defining classes of classes, or attaching properties to classes). In practise, complete OWL Full implementations do not currently exist. Usually nobody wants to develop an ontology by hand (one should keep in mind that the tools used have to provide the needed functionality).

This research project recommends the use of OWL DL sub-language for company internal ontologies with respect to the possible complexity and the guarantee for completing searches within the ontology. Simple public ontologies such as descriptions about the company's characteristic and products can use OWL Lite because of the reduced complexity and quantity.

For the complete OWL Web Ontology Language Reference see Dean & Schreiber, 2004.

Topic Maps

Topic Maps is standardised by ISO standard ISO/IEC 13250:2002 for representation and interchange of knowledge. XML encoding of topic maps is described as XML Topic Maps (XTM) [Pepper & Moore, 2001].

Building blocks of topic maps are: Topics, names, occurrences and associations [Rath, 2003].

A **topic** is a resource that represents a subject in a topic map. A subject is defined as general as possible: "In the most generic sense, a 'subject' is any thing whatsoever, regardless of whether it exists or has any other specific characteristics about which anything whatsoever may be asserted by any means whatsoever." [ISO Topic Maps, 2002] This allows the use of every conceivable domain.

Topics may have zero or more **names**. Each name is valid within a certain scope, for example for a certain language in an international (multi-lingual) topic map. The "Base Name" is a string used as topic name. In addition, a "Variant Name" can be used as an alternative of a base name optimised for special operations, such as sorting. Any kind of a resource is allowed (including a string). An example is given in Figure 2.2-9.

Occurrences are links to resources that are relevant to the topic. A resource can be referenced by a URI or described by a text.

Associations are used to describe relationships to other topics (and only topics).

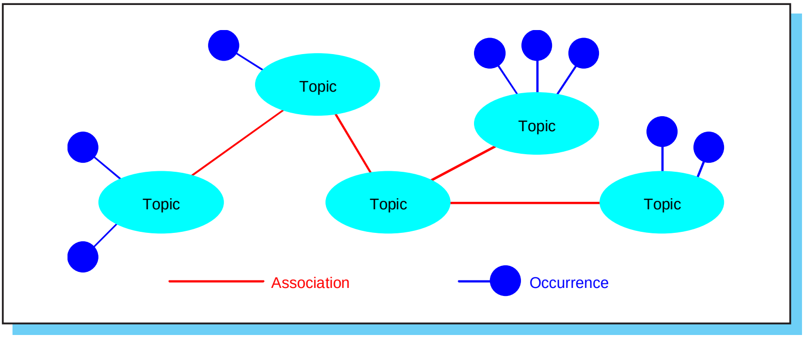

Figure 2.2-7 Example of a Topic Map

Figure 2.2-8 shows how resources can be referenced and how an association is defined. In that example the association between Robert and CIT has no name. Both topics are linked to a different home page but both are linked to the same document (in the middle).

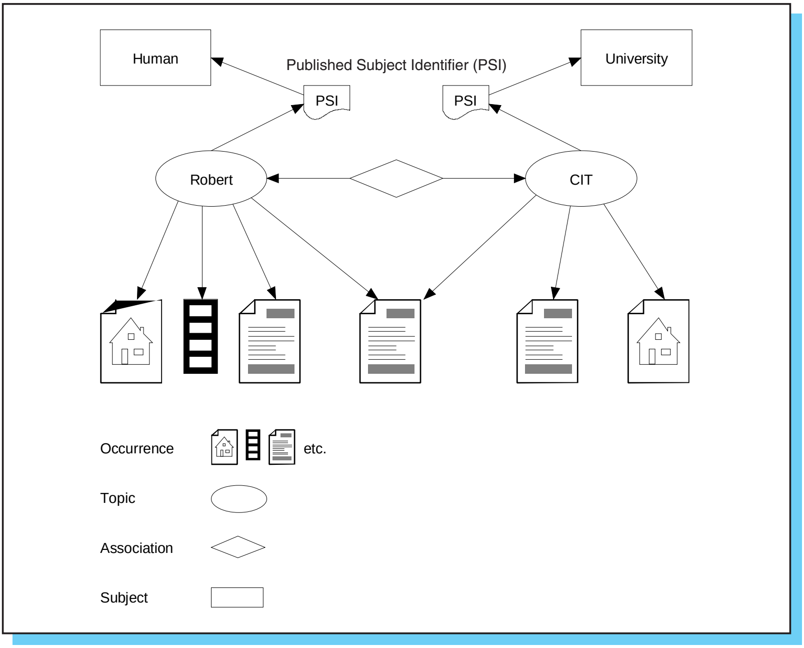

Figure 2.2-8 Topics, Occurrences and Associations in Topic Maps

For the identification of topics, a “Subject Identifier” (SI) is used as a unique ID. If a subject identifier is published for the use by other people or systems, it is called “Published Subject Identifier” (PSI). In Figure 2.2-9, a topic with the SI “rloew” is defined with two base names as synonyms (“Robert” and “Mr. Robert Loew”) and a variant for displaying (a picture referred by a xlink reference) and a variant for sorting as usual in Yellow Pages (loew, robert).

Names and occurrences of topics and associations can have validities within special conditions – this is supported using the **scope** approach. For example, by adding a scope in the code of the example shown in Figure 2.2-9 as done in Figure 2.2-10, an application is able to switch between a formal and an informal mode (for example, sending community emails using “Dear Robert” or emails from company management using “Dear Mr. Robert Loew”).

```
<topic id="rloew">
    <baseName>
        <baseNameString>Robert</baseNameString>
    </baseName>

    <baseName>
        <baseNameString>Mr. Robert Loew</baseNameString>
    </baseName>

    <variant>
        <parameters>
            <subjectIndicatorRef xlink:href=
            "http://loew-da.de/images/loew.jpg"/>
        </parameters>

        <variantName>
            <resourceData>
                loew,robert
            </resourceData>
        </variantName>
    </variant>
</topic>
```

Figure 2.2-9 Examples of Names in Topic Maps

```
<baseName>
    <scope><topicRef xlink:href="#informal"/></scope>
    <baseNameString>Robert</baseNameString>
</baseName>

<baseName>
    <scope><topicRef xlink:href="#formal"/></scope>
    <baseNameString>Mr. Robert Loew</baseNameString>
</baseName>
```

Figure 2.2-10 Example of Using Scopes in Topic Maps

Inheritance model

The Topic Map Inheritance model is described in Appendix B.2.

For further information, Rath provides a paper describing Topic Maps in detail in a practical scope [Rath, 2003].