

Frode Hjeltnes

Development of a Topic Map
Application for the Project QUIS
- QUality, Interoperability and
Standards in e-Learning

Trondheim, January 2006

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and
Electrical Engineering
Department of Computer and Information Science

Master's thesis

Programme of study: Master of Science in Computer Science

Supervisor: Arvid Staupe, IDI

Foreword

This report is the master thesis of Frode Hjeltnes. It is written at the Department of computer and information science, Faculty of information technology, mathematics and electrical engineering, Norwegian university of science and technology during a period of 20 weeks, from August 2005 to January 2006. The main objective of the project has been to develop an application to create and display information from a topic map.

The student is a member of the project “Quality, Interoperability and Standards in e-Learning” (QUIS), Work Package 5 – “Standards for e-Learning”. The results from the students’ work will be included in the QUIS report.

I would like to thank Arvid Staupe (mentor) for help and feedback. I also thank the other members of the QUIS project. Particularly Andrea Sterbini and the rest of work group 5, for being willing to postpone the deadline of Work Package 5. Finally I thank Geir Ivar Holberg for proofreading the report and helping out during testing.

Trondheim Jan 30th 2006.

Frode Hjeltnes

Summary

Quality, Interoperability and Standards in e-Learning (QUIS) is an EU funded project, and is a part of the e-Learning program¹. The projects contract number within the e-Learning program is 2004-3538/001-001 ELE-ELEB14. The project has partners from Norway, Sweden, Italy and Hungary. This thesis is written to be a part of the projects' Work Package 5 – “Standards for e-Learning”.

A topic map is an information structure for describing and identifying resources. It consists of the major elements topics, occurrences and associations. A topic is the representation of a real world subject. An occurrence is a source of information about the topic. Internal occurrences are information sources within the topic map file. External occurrences are links to resources located elsewhere. An association is a relationship between topics. It describes how the topics are related to each other, by giving the different members of the association “roles”.

All topic map elements can be scoped to give them extra semantic value. Two or more topic maps can be merged to form one large topic map. Topic maps usually follow the XML for Topic Maps (XTM) standard.

In addition to being an information structure, the topic map standard also tries to overcome the so called “Web Identity Crisis”. The Web Identity Crisis is based on the statement that “there is no way of knowing what references on the World Wide Web identify today”.

The Semantic Web is a similar technology to topic maps. It is based on the Resource Description Framework.

QUIS topic map is a topic map made to represent subjects in the area of e-Learning standardization. It is made in a “topic map constructor system” based on the Topic Map 4 Java API. A “topic map application system” can be used to display the content of the topic map. It is shown that this approach offers easy and structural access to subjects within this area of interest. In comparison with regular web portals, the topic map application has several advantages:

- The connection between related information is more intuitive.

¹ <http://www.elearningeuropa.info/>

- The topic map is easier to maintain extend.
- The topic map has better support for multiple languages.
- The topic map can be merged with other topic maps.

In order to exploit the full strengths of the topic map standard, more e-Learning communities have to take it into use. The creation of Published Subject Identifiers (PSI) will advance this process.

Table of contents

1	Introduction	1
1.1	Problems.....	1
1.1.1	Problems with search engines	1
1.1.2	Problems with web portals	2
1.1.3	How topic maps might solve the problems	2
1.2	Assignment description	4
1.3	The layout of the report.....	4
2	Theory	6
2.1	Topics	6
2.2	Topic types	7
2.3	Occurrences	8
2.4	Associations	10
2.5	Scopes.....	12
2.6	Published subject indicators (PSI).....	13
2.7	Merging	14
2.8	Navigational structure – topic map ontology	15
2.9	XML for topic maps (XTM)	17
2.10	Semantic web and topic maps	17
2.10.1	The Semantic Web	18
2.10.2	Resource Description Framework (RDF)	19
3	Pre study.....	20
3.1	Requirements for possible solutions	20
3.1.1	Finalization of the solution must be manageable within 20 weeks.....	20
3.1.2	Student centered development	20
3.1.3	Minimal development cost.....	20
3.2	Existing commercial topic map software	21
3.2.1	Ontopia.....	21
3.2.2	Infloom.....	22
3.2.3	Mondeca.....	22
3.2.4	Empolis.....	22
3.2.5	NetworkedPlanet	23
3.3	Existing open source topic map software.....	23

3.3.1	TMAPI	23
3.3.2	TM4J	24
3.3.3	TinyTIM.....	24
3.4	Developing software from scratch	24
3.5	Evaluation of alternative solutions	25
3.6	Available tools and technologies.....	30
3.6.1	PHP.....	30
3.6.2	JSP.....	30
3.6.3	Servlets	31
3.6.4	Tomcat.....	31
3.6.5	Cocoon	32
3.6.6	Evaluation of tools and technologies	32
3.7	Selection of development model.....	33
3.7.1	eXtreme Programming (XP)	33
3.7.2	Prototyping	33
3.7.3	Spiral development model.....	33
3.7.4	Waterfall.....	33
3.7.5	Looping waterfall model	34
3.7.6	Choice of development model.....	34
4	Requirements specification document	35
4.1	The organization of the requirements specification document	35
4.2	Non functional requirements	38
4.2.1	Implementation.....	38
4.2.2	Software	40
4.2.3	Accessibility	40
4.2.4	Security.....	41
4.2.5	User documentation.....	42
4.2.6	System documentation	42
4.2.7	Code documentation.....	43
4.2.8	Test documentation	44
4.3	Functional requirements	45
4.3.1	Functional requirements for the topic map creator system (system1)	45
4.3.2	Functional requirements for the topic map application system (system2).....	46
4.4	Requirements to the graphical user interface	49

4.5	Summary of the requirements	50
4.6	Superior test plan.....	52
4.6.1	Tests to be carried out	53
4.6.2	Check lists	53
4.6.3	Test specifications	53
4.6.4	Black box and white box testing	54
4.6.5	Test evaluation	54
5	System design document.....	55
5.1	Superior system description	56
5.1.1	System description	56
5.1.2	System modules.....	57
The client module.....		57
The server/JSP module.....		57
The java module.....		58
The topic map.....		58
The topic map constructor module.....		58
5.2	The server/JSP module.....	59
5.2.1	Web application server.....	59
5.2.2	The JSP module.....	60
Page description		61
5.3	The Java module.....	63
5.4	The topic map constructor module.....	65
6	Implementation and documentation	70
6.1	Programming code examples	70
6.1.1	Topic map constructor code example.....	70
6.1.2	Topic map application system code example.....	75
6.1.3	JSP page code example	77
6.2	Installation Guide	79
6.2.1	Topic map constructor system.....	79
6.2.2	Topic map application system.....	79
Installing Tomcat.....		79
Installing the JSP and Java files		80
6.3	User's Guide.....	81
6.3.1	Starting the system	81

6.3.2	Importing or creating topic maps	81
6.3.3	Select element to be added	81
6.3.4	Add topic	81
6.3.5	Add association	82
6.3.6	Add occurrence	83
7	Test document	85
7.1	Test principles and methods	86
7.1.1	“White box” testing	86
	Static analysis	86
	Dynamic analysis	87
7.1.2	“Black box” testing	87
7.1.3	Checklists	88
7.2	Tests	88
7.2.1	Testing requirements concerning documentation	88
7.2.2	Testing the graphical user interface	89
7.2.3	Testing requirements concerning implementation	90
7.2.4	Testing requirements concerning software, accessibility and security	90
7.2.5	Testing functional requirements for the topic map constructor system	91
7.2.6	Testing functional requirements for the topic map application system	91
7.3	Test results.....	92
7.4	Test conclusion.....	96
8	Results and discussion.....	98
8.1	Results from the topic map constructor system	98
8.2	Results from the topic map application system.....	99
8.2.1	Topic maps can improve web portals.....	99
8.2.2	Easy to maintain and extend.....	99
8.2.3	May use common interfaces.....	100
8.3	Discussion concerning TM4J	100
9	Further work.....	101
9.1	System enhancement	101
9.1.1	Topic map constructor system.....	101
9.1.2	Topic map application system.....	102
9.2	Additional work.....	103
9.2.1	Documentation	103

9.2.2 PSI	103
9.2.3 Open source publication.....	103
10 References	104
10.1 Topic maps	104
10.2 QUIS.....	104
10.3 TM4J	104
10.4 Technologies	105

List of figures

Figure 1 Quis Topic Map topics.....	7
Figure 2 Quis Topic Map Topic Types	8
Figure 3 Quis Topic Map Occurrences	9
Figure 4 Quis Topic Map Associations.....	10
Figure 5 Quis Topic Map Association Roles	11
Figure 6 Quis Topic Map Scopes	13
Figure 7 QUIIS Topic Map Navigational Structure	16
Figure 8 System overview	56
Figure 9 Superior sequential diagram	57
Figure 10 Server-Client block diagram.....	59
Figure 11 Sequential diagram for client-server communication.....	60
Figure 12 JSP page diagram.....	61
Figure 13 Java module class diagram	65
Figure 14 Topic map constructor system class diagram	66
Figure 15 Data flow diagram for the topic map constructor system.....	68
Figure 16 Select topic map element	81
Figure 17 Add Topic	82
Figure 18 Add association.....	83
Figure 19 Add Occurrence	84
Figure 20 Static analysis	86
Figure 21 "Black box" testing	88

List of tables

Table 1 Infoloom services and rates.....	22
Table 2 Evaluation of Ontopia Knowledge Suite.....	25
Table 3 Evaluation of Infoloom	26
Table 4 Evaluation of Mondeca	26
Table 5 Evaluation of Empolis.....	26
Table 6 Evaluation of NetworkedPlanet	27
Table 7 Evaluation of TMAPI.....	27
Table 8 Evaluation of TM4J.....	27
Table 9 Evaluation of TinyTim.....	28
Table 10 Evaluation of developing software from scratch.....	28
Table 11 Summary of evaluations.....	28
Table 12 Explanation of requirement IDs	36
Table 13 Explanation of importance estimation	37
Table 14 Summary of the system requirements	50

1 Introduction

The main objective of the thesis is to examine the use of topic maps, in order to find their strengths and weaknesses and to compare topic map applications with regular web portals. For this purpose a topic map application will be made, as well as a topic map within the area of e-Learning standardization. The application must be able to create a topic map according to the topic map standard, and display the information in the map.

Information retrieval from the Internet is usually done via web portals or search engines. There are several problems with these approaches, both for the users and the system administrators. This introduction will first describe these problems. It will then be presented how this project will use topic maps to see if they can solve the problems. Finally in the introduction, the complete assignment description and the layout for the rest of the report is described.

1.1 Problems

Finding information on the Internet today can be difficult. To find information within a specific area of interest, two common approaches are used:

- The use of search engines.
- The use of web portals

1.1.1 Problems with search engines

Search engines are huge databases with tables containing data about a large number of web pages. The search engines collect their data in different ways. What is common for practically all of them is that they spawn web pages and sort out words and phrases from the text and the page headers. When the user does his search, the engine presents the web pages containing the word or phrase searched for. The ordering of the pages presented depends on different parameters like:

- Where in the document the search word was found. A hit in the header gives for instance more credit to the document than a hit in the page text.

Introduction

- Number of links to this page from other pages.
- Number of links from this page to other pages of high rank.
- The page rank.

The problem with search engines is that the user will always be presented with an (most often) enormous amount of un-relevant search results, no matter how effective the search engine is. Identifying which pages are the ones of most interest can be a time consuming exercise.

1.1.2 Problems with web portals

The second approach for retrieving data is by using a web portal. A web portal is a web page which serves as a reference book for information, either within a specific area of interest or within several. A web portal within a specific area often contains lots of information. There are several problems with web portals. One of them is how to locate the information you're interested in. Information is often ordered in categories. For some kind of information the retrieval might be intuitive, for others not. This often depends on the user's experience with the web portal. When he has found the requested information, another question appears: How to find information related to the one just found? When investigating a specific area of interest the user is often interested in several sources. Only a small amount of these resources will fall under the same category or sub-category as the one being investigated at the present time. Finding the new information will therefore often take just as much time as finding the first information. The web portals often present a structure of links in order solve this problem, but they are rarely sufficient. The links are also hard to keep updated.

1.1.3 How topic maps might solve the problems

Topic maps are ontologies for describing and identifying resources. The topic map consists of topics representing subjects of interest. Information about the topics are both contained within the topic map, as well as linked to from the topic map. Information about how topics are related to each other is also present in the topic map. A thorough description of topic maps follows in chapter 2.

Introduction

The Work Package 5 (WP5) of the project QUality, Interoperability and Standards in e-Learning (QUIS), focus on the area “Standards for e-Learning”. The Work Package is chaired by the University of Rome: Universita’ di Roma “La Sapienza”. The results from the work in this Work Package are published on the projects’ web pages². The report contains information about e-Learning standards, learning management systems, publishing tool, digital repositories, etc.

It was a difficult task to collect all of the information written in the WP5 report. There exist huge amounts of data about e-Learning standardization on the Internet. It was hard to find the most relevant and most updated information. The members of work group 5 had to use search engines and web portals for this task. In addition to the web pages of major interested parties interested in e-Learning. Some of the actual web pages and web portals were:

- <http://www.estandard.no/> - A Norwegian web portal that deals with e-Learning
- <http://www.w3.org/> - The World Wide Web Consortium
- <http://www.ieee.org/> - The Institute of Electrical and Electronics Engineers, Inc
- <http://www.aicc.org/> - The Aviation Industry CBT Committee
- <http://www.adlnet.org/> - The Advanced Distributed Learning
- <http://www.imsglobal.org/> - The IMS Global Learning Consortium, Inc

It was decided within WP5 to develop a topic map application containing the information written in the project report. The objective of this was to be able to make a comparison between the topic map application and the existing web pages and portals concerning e-Learning standardization. The topic map application must therefore be available via web. In addition to finding out whether such an application provides easier access to information than search engines and web portals, it is also interesting to find out how time consuming it is to develop the application. The work will also tell something about the easiness of topic map maintenance and extension. The complete assignment description is shown in chapter 1.2.

² http://www2.tisip.no/quis/public_files/wp5-standards-for-elearning.pdf

1.2 Assignment description

“Development of a topic map application for the project QUIS – QUality, Interoperability and Standards in e-Learning.”

The project consists of programming a topic map application system for the project QUIS. The student must develop a system that provides the user with an easy and intuitive way of creating a topic map. The system must also allow the user to browse the topic map via a web browser. A server must be set up for this use.

The student must reason for his choice of programming environment

The topic map must follow the topic map standard: ISO/IEC 13250.

1.3 The layout of the report

This sub-chapter presents the layout for the rest of the thesis.

- **Chapter 2: “Theory”**
This chapter gives a thorough introduction to topic maps. It also presents some alternative technologies similar to topic maps.
- **Chapter 3: “Pre study”**
The pre study discuss’ strategies for system development, different development models, and different technologies that can be used.
- **Chapter 4: “Requirements specification document”**
The requirements specification document presents the system’s requirements. The requirements are the basis for the rest of the system development. The document can be seen as a contract of “what should be made”.
- **Chapter 5: “System design document”**

Introduction

The system design document presents how the system should be constructed, based on the requirements. It is meant as a basis for the actual implementation.

- **Chapter 6: “Implementation and documentation”**

This chapter presents examples from the programming code. It also contains an Installation Guide and a User’s Guide.

- **Chapter 7: “Test document”**

The test document describes how the system is tested. It presents the test principles that are followed, and the actual testing.

- **Chapter 8: “Results and discussion”**

In this chapter the results from the system development are presented. They are then evaluated with respect to the main objectives of the thesis.

- **Chapter 9: “Further work”**

This chapter discuss’ what further work can be done with the system after delivery.

2 Theory

This chapter gives a thorough introduction to topic maps. The different sub-chapters discuss' the following major elements and concepts in a topic map.

- Topics
- Topic types
- Occurrences
- Associations
- Scopes
- Published subject identifiers
- Merging
- Topic map ontology
- XML for topic maps (XTM)

During the discussion examples will be used to give a better understanding of the concepts. In some sub-chapters several examples will be presented. One example however will be used throughout the entire chapter. That is the QUIS topic map example, which is the topic map to be developed in this thesis.

At the end of the chapter an alternative informational structure, The Semantic Web is presented. The last sub-chapter discuss' the Resource Description Framework (RDF), which is the basis framework for the Semantic Web.

2.1 Topics

Topics are the main elements in topic maps. They represent the real world subjects of which the developer wish to present information. They can be of any existing or non-existing subject you can think of. The World can be a topic, just as a snow flake can be one. The opportunities are endless. More generally, every element in a topic map is a topic. See chapter 10 for references to supplementary information about topic map. For now, they are the representations of the subjects of interest. The topic map build in this project is in the area of e-Learning standardization. Topics in this area might be:

- Standard
- Standardization organization
- Publishing tool
- SCORM
- IEEE LOM
- Etc.

Figure 1 displays the topic map.

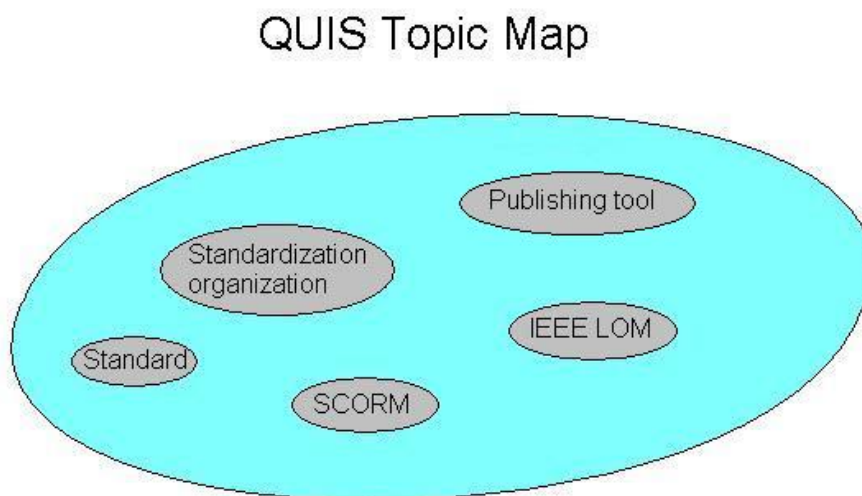


Figure 1 Quis Topic Map topics

2.2 Topic types

A topic map is ontology with inline structure both vertical and horizontal. A deeper explanation of these expressions follows in chapter 2.8. The vertical structure comes from typing the topics. This means to state that one topic is of the type of another one. For instance, you might have a topic “Vehicle”. If you then make another one, “Car”, this can be (or should be) stated as being of type vehicle. Further you can make a topic “Race car” which is of type car. This way you build a tree structure of the topics.

The typing of a topic is normally done when you add it to the topic map. It is not necessary to state that the topic which types the new one actually is a type. This comes from the definition of topic maps. If a topic is said to be of the type of another one, then the other one is by

default considered to be a topic type. When making applications for navigating and exploring the content of topic maps, this typing comes to use. It gives a navigational structure between the topics. Topic map applications can therefore display arbitrary topic maps with their correct structure, as long as they follow the topic map standard.

In the topic map developed in this project IEEE LOM is a standard. It is therefore written to be of the type “Standard”. The same applies for SCORM, even though SCORM in fact is a reference model containing a set of standards. If we add a couple of additional topics to the topic map, we get the following picture, displayed in Figure 2. The arrows point from the topic type to the topic being typed.

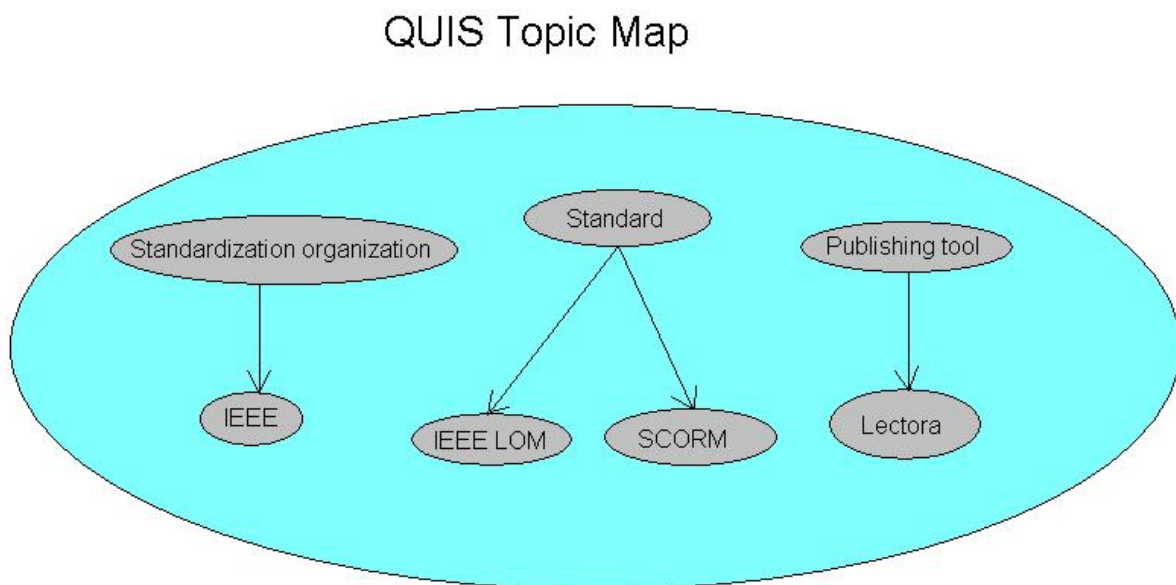


Figure 2 Quis Topic Map Topic Types

2.3 Occurrences

Topics are representations of real life subjects of which the developer wishes to present information. The actual information about the subjects is contained within the topics as so called occurrences. There are two types of occurrences; internal and external.

Internal occurrences are information about the subject that is typed inside the topic map. This information is included in the XTM file. In the QUIS topic map, an internal occurrence to the

Theory

topic SCORM might be “SCORM is a reference model developed by ADL.” This line of information tells us something about the topic SCORM. Internal occurrences have their own tag inside the topic map XTM file.

External occurrences are links to information about the topic, which is located elsewhere. Most typically these links are represented as URIs, and point to a web page. They may also point to a file in a file system, a field in a database, etc. Special kind of external occurrences are the subject identifiers. These are links where the file or the information it points to is the actual subject. If a topic is “NTNUs home web page”, the external occurrence “<http://www.ntnu.no/>” is such an identifier. Subject identifiers also have their own tags in the XTM file.

When adding an occurrence to a topic map topic, you have to specify whether it is an internal or an external occurrence. You must also specify of which type the occurrence is. That is; both internal and external occurrences have to be of a type, which again is a topic. Figure 3 shows the QUIS topic map, which now is extended to include some internal and external occurrences. The internal ones are written in blue, and the external ones in red. The topics which in our case are typing topics might of course have their own occurrences.

QUIS Topic Map

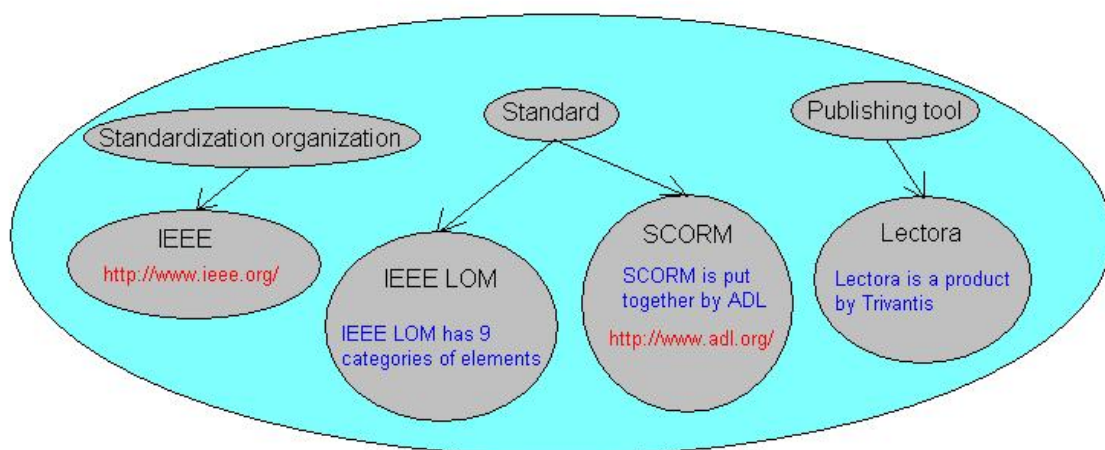


Figure 3 Quis Topic Map Occurrences

2.4 Associations

Associations are the last of the three major elements in topic maps – topics, occurrences and associations. Typing topics represents the vertical structure of the topic map ontology.

Associations bring the horizontal one. Associations say something about how topics are connected to one another. An association can be a connection between an arbitrary numbers of topics. It might even be between a topic and the same topic. In that case it describes how the topic is connected to itself. Further on in the thesis an association is considered to be between two topics. However, it could easily be extended to include associations of an arbitrary number of topics. How a topic is connected to another topic is to a certain degree included in the association. It is not only a link saying that there is a connection.

The topic “IEEE LOM” could for instance be a member of two different associations. In one of them the association is with the topic “IEEE”. In the second, the topic “SCORM” is the other member. Figure 4 shows these two associations at a basic level.

QUIS Topic Map Associations

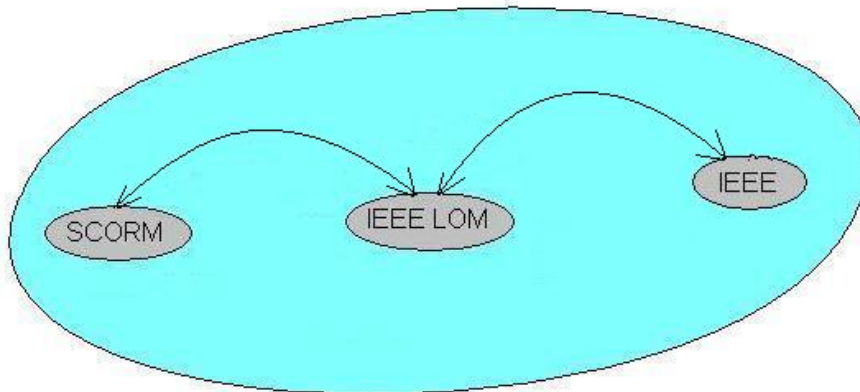


Figure 4 Quis Topic Map Associations

An association always goes both ways. It defines the topics as being dependent of each other. If one topic is associated with a second, the second is always associated with the first. This comes from the fact that a topics’ memberships in the associations is written in an association element, and not in the topic itself. Association elements have their own tags in the topic map. Within this XML structure you add members of the association. You can add as many members as you like, and therefore as many topics as you like. Each member in an association

is given a role. This provides extra semantics to how the topics are associated. A topic can have different roles the different associations where it is a member of.

In the QUIS topic map, the topic “IEEE LOM” has different roles in its two associations. In the association with “SCORM” it has the role “included in”. This means that IEEE LOM is a standard that is included in the SCORM reference model. “SCORM” has the role “includes”, which meaning should be obvious. In the association with “IEEE”, “IEEE LOM” has the role “developed by”, and “IEEE” has the role “has developed”. The meanings here should also be clear. Figure 5 is extended to include these roles.

QUIS Topic Map Associations

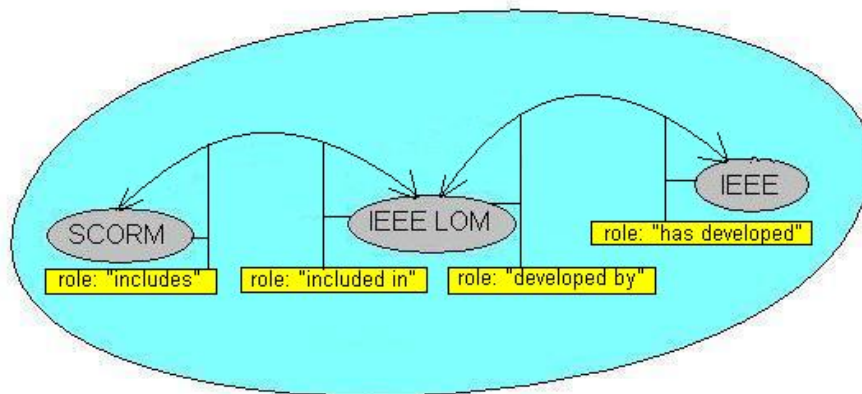


Figure 5 Quis Topic Map Association Roles

Each association is given a unique name and a type, the type being a topic. This way it is easy to retrieve all associations that are of the same type, and thus contains closely related information.

In the QUIS topic map, each standard that is a part of a reference model is a member of an association stating this fact. A topic map application should give the user the possibility to list all associations of this type, and with that, a complete overview over all the reference models and the standards they are built upon.

2.5 Scopes

A scope defines under what conditions the information in the actual topic map object is valid. This is useful when you build topic map applications for users with different backgrounds. As a user of the application you might choose which scope is applicable for you. Topics, associations and occurrences can be scoped.

Scopes can be used in the QUIS topic map. The QUIS topic map is based upon a report from work group 5 – Standards for e-Learning. The information in this report is useful for all people interested in e-Learning. It is also desirable to present information about the QUIS project in the topic map, in addition to information about e-Learning standardization. There will be added topics related to QUIS. Some of this information however, is not meant for all and everyone to read. This could be personal information about the project members, or internal drafts. Today the project has a web page at <http://www.tisip.no/quis>. At this page there is a link to “Partners’ arena”. To go to this page you need a password. Password protection can be integrated in a topic map application using scopes. A link for partners’ arena might be added and people who have the password would be allowed to enter. Here extra information about the QUIS topics that are scoped “partners arena” will be shown. Figure 6 shows some QUIS topics, with scoped occurrences.

Another example might be a topic map containing topics representing learning objects for educational purpose, the scopes might be the different levels of education. The theory of relativity might be a topic. A vast number of learning objects could be written about this subject. For a student in high-school, the most advanced ones would be too difficult, whereas the easiest ones would be of no interest to a doctor graduate. Using scopes you can build an application where the end user may filter the information and end up with the topic information most likely to be of interest for him, just by choosing a specific scope. Such an application might of course use another name than scope, for instance “level of education”.

Scopes are also convenient when you merge two or more topic maps. Merging will be discussed more thoroughly in chapter 2.7.

QUIS Topic Map Scopes

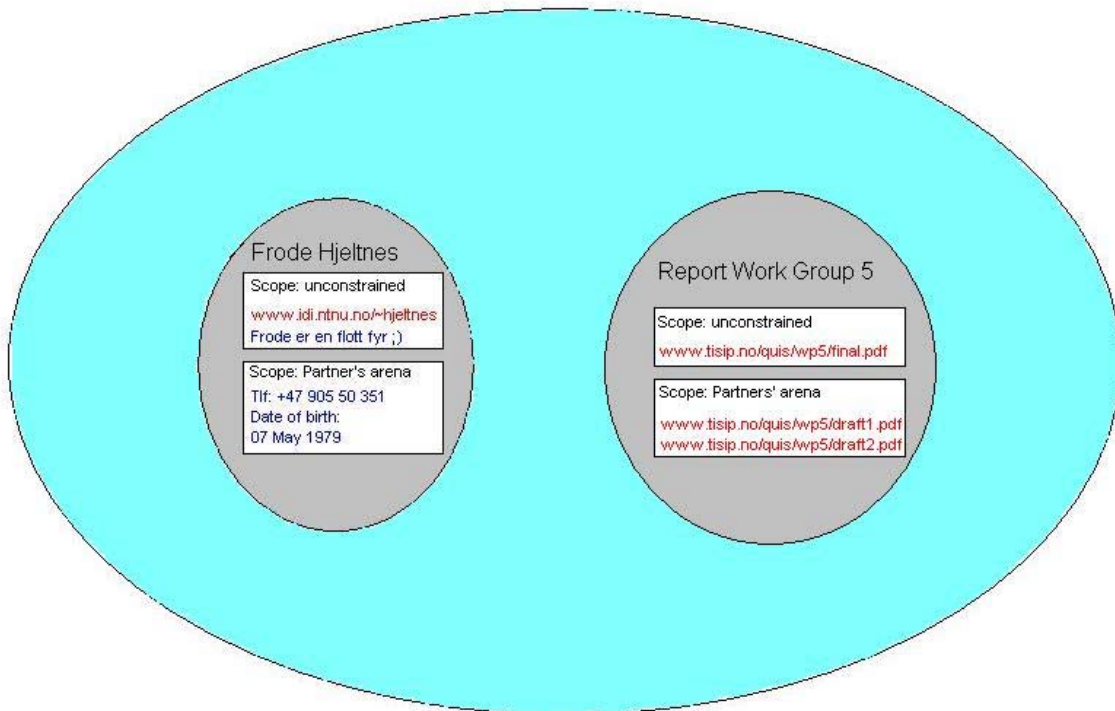


Figure 6 Quis Topic Map Scopes

2.6 Published subject indicators (PSI)

All topics must have a unique identifier within the topic map. Two topics can not have the same identifier. If they do they are considered to be the same topic. A way of identifying a topic is by using a PSI. PSIs are usually URIs leading to a web page defining the topics.

Using PSIs has no advantage over the use of topic identifiers, within a specific topic map. The situation is another if you search in multiple topic maps simultaneously, or if you merge topic maps.

If for instance another organization or project group also is developing a topic map in the area of e-Learning standardization, it would be useful to merge these two maps into one big one. This will lead to a single topic map containing all the information from both ones.

A topic map in this area would most likely contain a topic called “Standard”, since standards are most important in e-Learning standardization. When you merge the two maps you would

only like one such topic in the final topic map. If the identifiers for “Standard” in both topic maps are the same, this result will be obtained. If they are identified differently, you would have two topics about the same subject in the final map. “Standard” might for instance be identified as “e-Learning standard” in one topic map, and just “Standard” in the other one. Since there is no way of knowing how other people identify their topics, this would most likely be the case for a lot of topics.

The use of PSIs solves this problem. A public web page with PSIs for e-Learning standardization could be made. These PSIs could then be used in addition to internal topic identifiers within each topic map. Two topics with different internal identifiers are considered the same when merging, if they have the same PSI. The result of merging the two maps would be one topic with two different names. There already exist web pages with PSIs in the area of languages and countries. There is expected to be made others in the near future.

2.7 Merging

One of the major qualities of topic maps is that they can be merged. The result of merging two topic maps is a single topic map containing all the topics in the initial ones. This functionality might be used to separate the burden of populating a large topic map between different persons working on the same topic map. It is even more useful when different persons or organizations have developed separate topic maps within the same area of interest. As long as both topic maps follow the topic map standard it is possible to merge the maps. The use of published subject indicators prevents the problem of duplicate topics representing the same subject. A topic that is present in both topic maps will appear as one in the new topic map, which will contain all associations and occurrences from both topic maps. By using scopes it is easy to make a multilingual topic map.

The topic map which will be made in this project will be in the area of e-Learning standardization. It will be written in English. Although the author is not at the present time aware of any other published topic map in this area, it is not unlikely that one will be made in the near future. Both e-Learning standardization and topic maps are areas in growth.

If for instance a topic map written in Norwegian within this area is published, it would be preferable to merge the maps. This is easily done if both maps used the same PSIs. All

elements in the Norwegian topic map should be scoped “Norwegian” and all elements in the QUIS topic map should be scoped “English”. A topic map application supporting scopes will then give the user the possibility to choose either Norwegian or English. In this way large portals can be made, presenting information about e-Learning standardization, and the portals can also be common for people with different native languages. Easy access to relevant information is one of the main purposes for developing the topic map standard in the first place. And it is exactly what will be achieved by merging several topic maps within the same area of interest and presenting the map at a portal common for all.

2.8 Navigational structure – topic map ontology

Phrases like “vertical structure” and “horizontal structure” has been used in the previous sub-chapters. The meaning of this will be explained here. Most users of a topic map are interested in information about the subjects represented by the topics. Users may also want to investigate topics related to or in the same area as the one they are looking at. This section explains how a user navigates within the topic map to find this new information?

If the topic map application offers search functionality it is of course possible to use this. It can be quite time consuming to do this, and not always intuitive to know what to search for. A topic map offers two easy ways of navigating to closely related information within the topic map. In this report they have been called vertical and horizontal.

As explained earlier topic typing gives the vertical structure. A topic map application should present the topic type of each topic. In addition to that it should also present the topics that are typed by this topic. This information is a part of the topic map itself and can easily be extracted and presented to the user. This is illustrated in Figure 7 with arrows pointing up and down.

These topics have a “parent – child” relationship, normally presented as arrows pointing up and down in a tree graph. The topics’ parent and children is often closely related to the topic itself, and might provide interesting information to the user. If for instance a topic is “car”, the topic type might be “vehicle”. Topics typed by “car” might be specific cars or another topic type like for instance “race car” or “remote controlled car”. In the QUIS topic map

“publishing tool” might be a topic. Its topic type might be “e-Learning software” and topics of this type might be “Lectora” and “Macromedia Authorware”.

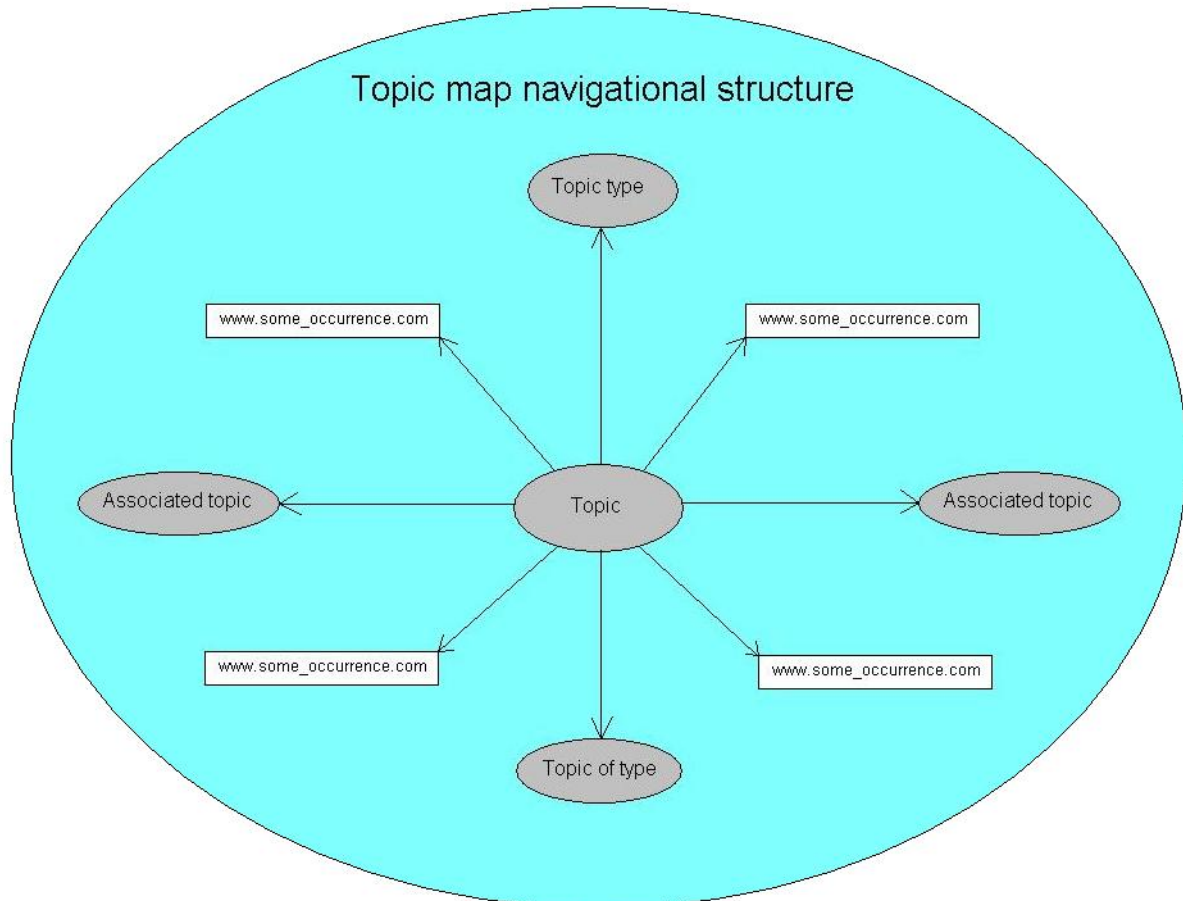


Figure 7 QUIS Topic Map Navigational Structure

Provided that the topic map application offers the most common navigational functionality, how could a user find related information when he is finished investigating the topic called “topic” in the figure? The answer is by the use of associations. They provide the “horizontal structure”, and can lead to related topics. Within the association there is information about how the topics are related, i.e. what roles they play in the relationship. The association has a name which says something about the nature of the relationship. All this information is extracted directly from the XTM code, without the web editor having to do any extra work. The grouping and presentation of associated information in a normal web portal is very often poor and without consistency. It is at least not consistent between different web portals. When moving from one web portal to another, the user must very often adapt to a whole other interface and concept of navigation. Different topic map applications may have different

layout, but the navigational structure will most likely remain the same. It comes more or less from the standard itself. Navigation via associations is illustrated in the Figure 7 by arrows pointing to the left and to the right. The topic “car” might be associated with the topics “boat” and “aircraft”, which could be interesting for users interested in cars. The topic “e-Learning software” might be associated with the topic “e-Learning software vendors”. The opportunities are endless.

Both this vertical and horizontal structure is often attempted implemented in web portals, but the actual solution is not always attractive. Extending the structure with new information also takes a lot of time because new web pages have to be made manually. This is not the case in topic maps. Adding new topics and/or associations or typing some is easily done. The topic map application then only has to reload the topic map, and the new relationships will be present.

The arrows pointing to the boxes with web addresses represent external occurrences. This information is also present within the topic map XTM file. The links does however lead to information outside the topic map and is therefore not considered a part of the topic map navigational structure (at least not in this paper).

2.9 XML for topic maps (XTM)

XTM is a XML schema used for topic maps. It contains tags for all the topic map elements. The structure and different tags in the XTM standard will not be further discussed in this report. More information about the standard can be read at³. The complete standard is also added as Appendix A.

2.10 Semantic web and topic maps

The semantic web is a framework developed by the World Wide Web Consortium (W3C)⁴. It is described in the following way at their own web pages:

*“The **Semantic Web** provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by*

³ <http://www.topicmaps.org/xtm/>

⁴ <http://www.w3.org/>

W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.”

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.”

The Semantic Web can in certain areas be an alternative to topic maps. This chapter first describes the Semantic Web more in detail. The RDF framework is presented, and finally a comparison between RDF and topic maps is presented.

2.10.1 The Semantic Web

The Semantic Web is a project that aims at providing web resources with additional information. This information can be used by software agents (i.e. web browsers) to do automatic reasoning on behalf of the user.

Today a web browser knows only how to draw a web page based on tags and links. There is no way a web browser can do some reasoning on its own, since there exist no precise and consistent definition of how subjects and information is identified on the Internet. This fact is called “the Internet identity crisis”. Identification today is most often done using URIs. There exists however no defined or standardized way of interpreting the information in a URI or the information it leads to. The URI <http://stud.ntnu.no/~hjeltnes> points to the homepage of Frode Hjeltnes. There is no way of telling whether it identifies Frode Hjeltnes, Frode Hjeltnes’ homepage or neither. Before this identity crisis is solved it is not possible to implement a semantic web that enables software agents to perform actions and reason based on the webs’ structure and content.

The following case exemplifies how the Semantic Web might do reasoning of its own. Let’s say a person is abroad and wants to rent a car. Today the simplicity of this task depends on the persons’ location. With the Semantic Web he can tell the web browser when, where and for how long he wishes to rent the car. In addition to a maximum price he is willing to pay for it. The new and “intelligent” web browsers can then locate car rental agencies in the area, and examine whether they have available cars for the requested period of time and within the price

limits. Further on can they for instance check against a database weather potential car rental agencies satisfies certain requirements. The web browser may also have access to the users' bank account and pay for the deal. The car rental agency's web browser might check weather the customer has a valid drivers license, and perhaps even if he's got a record as a car thief.

To realize this kind of web browsers an information structure other than HTML is needed. This new standard is the Resource Description Framework (RDF). It is described in chapter 2.10.2.

2.10.2 Resource Description Framework (RDF)

RDF is the standard that is meant to be the fundament of the development of the Semantic Web.

RDF is a framework that describes web resources. The formal name of the framework is "The original RDF Model and Syntax Recommendation", and it is developed by The RDF Core Working Group⁵. The framework is written in XML, and a vocabulary for resource description is made along with the framework.

⁵ <http://www.w3.org/2001/sw/RDFCore/>

3 Pre study

In the pre study the alternative approaches for creating the application is considered. The different possible solutions will be compared with three superior requirements for the system development. These requirements will be presented prior to the alternative solutions. When a solution is chosen, different technologies that might be used are discussed. The technologies will also be graded with respect to how proper they seem for the chosen solution. Some technologies are then chosen for use in the development, based on the grading. Details on how these technologies will be used in design and implementation does not belong in the pre study. Finally some system development models are discussed, and one is chosen for use in the remaining phases of this project.

3.1 Requirements for possible solutions

Three superior requirements exist for choosing what type of development that will be done.

3.1.1 Finalization of the solution must be manageable within 20 weeks.

A master thesis at NTNU is scheduled for 20 weeks of work. Possible solutions must therefore be adaptable to fit in to this time limit.

3.1.2 Student centered development

This is a single student thesis with focus on research and system development. Using “on-the-shelf” software is therefore only applicable if it’s a pre-requisite for further research and programming.

3.1.3 Minimal development cost

A master thesis has no budget what so ever. As this thesis is written as a part of the QUIS project, minor development costs up 1000€ is acceptable.

As a result of these requirements three possible solutions appear:

- To use an existing commercial topic map software.
- To use an existing open source topic map software.
- To develop the entire system from scratch.

3.2 Existing commercial topic map software

One possibility for developing a topic map is to use the software and/or consulting services of existing topic map vendors. According to the leading topic map web page⁶, the five major topic map vendors are:

Ontopia⁷

Infoloom⁸

Mondeca⁹

Empolis¹⁰

Networked Planet¹¹

Chapter 3.2.1 to 3.2.5 describe their how they can be used in the system development.

3.2.1 Ontopia

Ontopia is said to be the worlds leading provider of generic topic map software. Their main products are the Ontopia Knowledge Suite (™) and the Ontopia Topic Map Navigator. The Knowledge Suite is a topic map engine and the Navigator is a framework for building web-based topic map delivery applications. For use in this project, licenses for the Knowledge Suite and the Navigator have to be bought. The main area of work would then be to use the Navigator to develop a topic map application which satisfies the project requirements. Further inquiries as to whether this would be suitable for this project is canceled, due to license prices. A single user license for the Knowledge Suite and the Navigator costs approximately 33.000\$, a price that is far above this projects' budget limit.

⁶ <http://www.topicmap.com/>

⁷ <http://www.ontopia.net/>

⁸ <http://www.infoloom.com/>

⁹ <http://www.mondeca.com/>

¹⁰ <http://www.empolis.com/>

¹¹ <http://www.networkedplanet.com/>

3.2.2 Infoloom

Infoloom base its business on consulting, helping others to develop topic maps. Unlike Ontopia, Infoloom does not offer a topic map engine as one of their products. They have developed a topic map engine and a developing interface called Topic Map Loom Technology. This however *“is used internally by infoloom.com to perform the work; it is not a part of any deliverable”*.¹² Infolooms’ services is based on a per hour consulting service with pre set rates. The services are divided into three different categories. The categories and their hourly rates are shown in Table 1.

Table 1 Infoloom services and rates

Service category	Hourly rate
Topic map production setup requirements analysis	\$250
Topic map production service	\$200
Topic map production	\$150

Since Infolooms’ services do not include a topic map engine, they are not of interest for this project. The hourly rates are also well beyond the acceptable price limit.

3.2.3 Mondeca

“Mondeca designs software components and solutions for index management enabling content organization, information retrieval and information visualization.”¹³ The student did several attempts to contact Mondeca. They did not reply, and their services will therefore not be considered.

3.2.4 Empolis

"Empolis - The Information Logistics Company - is the leading supplier of enterprise content and knowledge management solutions. Over 18 years of experience and comprehensive know-how, including substantial knowledge of complex business processes, build the foundation of our specific solutions designed for a multitude of diverse areas of application. Empolis has a proven track record, not only in using open industry standards, but also being in the driver's seat of developing and approving various new standards like ISO Topic Maps,

¹² <http://www.infoloom.com/services.htm>

¹³ <http://www.topicmap.com/topicmap/vendors.html>

TMQL or XTM."¹⁴ The student did several attempts to contact Empolis. They did not reply, and their services will therefore not be considered.

3.2.5 NetworkedPlanet

"NetworkedPlanet is a company of high-calibre topic map experts with extensive experience in developing and working with the topic maps standard. NetworkedPlanet was formed in 2004 to develop topic map processing tools and applications specifically for the Microsoft .NET platform... The company's first product... TMCORE05 has been designed as a toolkit to enable developers on the Microsoft .NET platform to create scalable topic map applications with robust persistent storage and full support for a multi-user environment. The engine supports the full range of the ISO standard Topic Maps specification and also provides a flexible web services API and easily-processed XML syntax to enable rapid application development and flexible deployment in a .NET environment."¹⁴ The student did several attempts to contact NetworkedPlanet. They did not reply, and their services will therefore not be considered.

3.3 Existing open source topic map software

A few attempts to develop open source topic map software exist. The two which have received some degree of public acceptance are Topic Map for Java (TM4J) and TinyTIM. They are both made to support the Topic Map Application Programming Interface (TMAPI).

3.3.1 TMAPI

TMAPI is a programming interface for developing topic maps in Java. TMAPI can be downloaded in a .jar-package from the developers' homepage.¹⁵ Its purpose is to serve as a standard interface for topic map developers. Without such a standard there will be no consistency between the different topic map applications. Independent developers always have different interpretations on how to best develop applications. This API is not a program. It is merely a set of interfaces describing how to represent topic map information. A standardized way of doing this will lead to programmers writing portable code, as well as a basis for programmers to communicate experiences on topic map programming. Current implementations of TMAPI are:

¹⁴ <http://www.topicmap.com/topicmap/vendors.html>

¹⁵ <http://www.tmap.org/>

- TM4J and TinyTIM, which will be described in the following two sub-chapters.
- XTM4XMLDB which is “an Open Source TMAPI implementation for native XML databases like eXist or Apache Xindice as backend.
- Ontopia Knowledge Suite which was described in Chapter 3.2.1.

3.3.2 TM4J

TM4J¹⁶ *“is an Open Source Java library for working with topic maps. It provides its own API, but also provides TMAPI interfaces. TM4J is a larger library than the pure TMAPI. It consists of four major parts:*

TM4J Engine is an engine for processing topic maps

TMNav is a graphical desktop application for browsing topic maps

Panckoucke is a library for creating abstract graph representations for topic maps

TM4Web supports integrating the TM4J Engine with commonly used web application frameworks

Along with the library come an Installation Guide, a Developer’s Guide and full API Documentation. TM4J supports three backend storage mechanisms; in-memory, Ozone (object-oriented database) and Hibarnate (relational database).

3.3.3 TinyTIM

TinyTIM is an Open Source TMAPI in memory implementation. It offers a few more functions, but supporting TMAPI is its major purpose.

3.4 *Developing software from scratch*

The last alternative for developing the QUIS Topic Map is to do it all from scratch. Whether this is manageable during 20 weeks is uncertain. This option will only be considered if no other solution is possible.

¹⁶ <http://www.tm4j.org/>

3.5 Evaluation of alternative solutions

In this sub-chapter the alternative solutions will be more thoroughly evaluated. At the end a final selection will be made as to which will be the solution of choice for the QUIS topic map. Each of the solutions within the categories; “existing commercial topic map software”, “existing open source topic map software” and “developing software from scratch” will be evaluated against the three overall requirements:

- R1: 20 weeks time limit
- R2: Student centered development
- R3: 1000€price limit

The different solutions will be rated at a scale from 0 to 3 for each requirement. The scale has the following interpretation:

- 0: The solution is not acceptable
- 1: The solution is possible
- 2: The solution is good
- 3: The solution is optimal

The different requirements have an equal weight. The total score is a multiplication of the different scores. This assures that a zero in any of the scores leads to a total score of zero. This is necessary since the definition of zero is that the solution is not acceptable. The solution with the highest total score is considered the best one, and will be chosen. Each evaluation is shown in a table. Finally the different evaluations are compared in a final table.

Table 2 Evaluation of Ontopia Knowledge Suite

	Score	Evaluation
R1	3	With the use of existing software and possible help from Ontopia, an acceptable solution could be achieved within 20 weeks.
R2	1	The student would be very dependent on the Ontopia software and staff.
R3	0	Due to the prices of this software, the solution is not

Theory

		acceptable.
Total score	0	The solution is not acceptable.

Table 3 Evaluation of Infoloom

	Score	Evaluation
R1	3	With the use of existing software and possible help from Infoloom, an acceptable solution could be achieved within 20 weeks.
R2	1	The student would be very dependent on the Infoloom software and staff.
R3	0	Due to the prices of this software, shown in Table 1, the solution is not acceptable.
Total score	0	The solution is not acceptable.

Table 4 Evaluation of Mondeca

	Score	Evaluation
R1	-	Mondeca did not answer upon request, and the use of their software is therefore not an option.
R2	-	
R3	-	
Total score	-	The solution is not acceptable.

Table 5 Evaluation of Empolis

	Score	Evaluation
R1	-	Empolis did not answer upon request, and the use of their software is therefore not an option.
R2	-	
R3	-	
Total score	-	The solution is not acceptable.

Table 6 Evaluation of NetworkedPlanet

	Score	Evaluation
R1	-	NetworkedPlanet did not answer upon request, and the use of their software is therefore not an option.
R2	-	
R3	-	
Total score	-	The solution is not acceptable.

Table 7 Evaluation of TMAPI

	Score	Evaluation
R1	1	The TMAPI is solely an application programming interface. It contains no written code. By just using TMAPI the situation will be much like programming the whole system from scratch. It is not likely that a functional system can be made within the time limits.
R2	3	This approach will not request involvement from other people than the student.
R3	3	The TMAPI is open source, and free of charge.
Total score	9	

Table 8 Evaluation of TM4J

	Score	Evaluation
R1	3	TM4J exists of a library of pre-programmed Java classes. The library is quite extensive and finalization within 20 weeks should therefore be manageable.
R2	3	The library does not contain any actual implementation of a topic map application. System development will therefore be solely student centered.
R3	3	TM4J is open source and free of charge.
Total score	27	

Table 9 Evaluation of TinyTim

	Score	Evaluation
R1	2	TinyTim contains a library of classes, much like TM4J. The library is however less extensive, and finalization within 20 weeks is therefore less likely than with TM4J.
R2	3	The library does not contain any actual implementation of a topic map application. System development will therefore be solely student centered.
R3	3	TinyTim is open source and free of charge.
Total score	18	

Table 10 Evaluation of developing software from scratch

	Score	Evaluation
R1	1	Developing an entire system from scratch might be possible. It is however not likely that this system will perform as well as systems made with basis in a pre-programmed library.
R2	3	This approach of course receives the highest score in this category.
R3	3	Developing a system from scratch is free.
Total score	9	

Table 11 Summary of evaluations

	R1	R2	R3	Total score
Commercial software				
Ontopia Knowledge Suite	3	1	0	0
Infoloom	3	1	0	0
Mondeca	-	-	-	-
Empolis	-	-	-	-
NetworkedPlanet	-	-	-	-
Open source software				
TMAPI	1	3	3	9

Theory

TM4J	3	3	3	27
TinyTim	2	3	3	18
Self development				
Developing system from scratch	1	3	3	9

The evaluation shows that using commercial software is far too expensive and therefore not an option. It also shows that some topic map vendors are not interested in requests made from a student.

The last solution, developing the entire system from scratch, is considered to be possible. Programming such a system within the time limit would however most likely not lead to a very good result. A master thesis at NTNU must contain much more than program code. The actual time set of for programming is approximately five weeks. The alternative is therefore not a good one. It is however possible, since a master thesis is equally based on research as on results.

The solutions receiving the highest scores are the ones which include using an open source library. Developing an application based on TM4J is the solution that receives the highest score. The reason for this is that this API is bigger and offers more functionality than the other ones.

The system will be made based on the TM4J library. The remaining part of the pre study discuss' the different technologies that can be used during the system development.

3.6 Available tools and technologies

So far TM4J has been chosen as a basis for the system development. It has not yet been discussed what other tools and technologies that can be used along with TM4J. The objective of this sub-chapter is to discuss some chosen solutions that are available and may be used in this project.

3.6.1 PHP

PHP is a recursive acronym for “Hypertext Preprocessor”. It is commonly used all over the World for web development, and can easily be integrated in HTML.

Usage in this project
Since the topic map application must be available on web, PHP is a feasible alternative as a programming and script language. It is flexible and quite easily used.

3.6.2 JSP

Java Server Pages is a dynamical extension of common web pages. It is a script language that opens for direct Java method calls in the web pages.

Usage in this project
JSP is quite similar to PHP. Since TM4J is a Java library, JSP seems like a better alternative. JSP also work well together with Tomcat, if Tomcat is to be chosen as web application server. The disadvantages with both JSP and PHP are the mix of logic and layout that appears when code is mixed within HTML.

3.6.3 Servlets

A servlet is in many ways an applet that is run on a web server. It is a Java technology that can be used to extend the functionality of web applications. Servlets are both server and platform independent, and have access to the entire Java API. They are written as common Java applications.

Usage in this project
The use of servlets is definitely a good alternative to make interactive web solutions. Tomcat supports servlets as well as JSP. By the use of servlets, the application can be written in Java with the entire Java API available.

3.6.4 Tomcat

Tomcat¹⁷ is a web application server written in Java. It is a reference implementation of Java Server Pages and Java Servlet, which are technologies from Sun Microsystems¹⁸. Tomcat is a servlet container. This means that it can run Java servlets on the server side of the application.

The difference between JSP and servlets when used in a web application is that JSP code (Java) can appear within the HTML document, while servlets are written as regular Java applications. It is common that JSP pages use the functionality within servlets by calling Java methods and then present HTML pages to the end user.

By the use of Tomcat as web application server, the program may be written in Java, and be placed in the container as web applications.

Usage in this project

¹⁷ <http://jakarta.apache.org/tomcat/>

¹⁸ <http://www.sun.com/>

Tomcat is an excellent choice for this system development. It is like written in Java like TM4J and can run servlets and JSP applications.

3.6.5 Cocoon

Cocoon¹⁹ is a XML and web development framework that can be used along with Tomcat. The framework contains generators and transformers for XML, and is build around the concepts of component based web development. It works like an assembly line where different components can be put together without much extra programming.

Usage in this project

Cocoon would be a useful tool for web development. Especially since it can be run as an independent servlet within Tomcat. Cocoon is however a complex tool and the student has little experience with using it.

3.6.6 Evaluation of tools and technologies

These tools and technologies were discussed because they all might be used in this system development. PHP and JSP are similar technologies where only one needs to be used. JSP is more suited since the rest of the discussed technologies also are written in or a part of Java. Cocoon was strongly recommended by experts at the institute. The student does however not feel that there is sufficient time to master the tool. It will therefore not be used.

The conclusion is that the following system development will take use of JSP, servlets and Tomcat.

¹⁹ <http://cocoon.apache.org/2.1/>

3.7 Selection of development model

There exist several different strategies in software development. Before any further work with the system can be done, a strategy or development model has to be chosen.

3.7.1 eXtreme Programming (XP)

eXtreme Programming means to begin with the code right away. The programmer is often only provided with input data and requested output data to/from the system. This is not a common strategy for a system as a whole, but rather for small units or modules of a system. Due to the lack of structure in this strategy it will not be used in this system development.

3.7.2 Prototyping

Prototyping means to quickly make a first version of the system. It might be a code implementation where some of the functionality is left out, or it might even be a sketch on a piece of paper. The first version is then presented to the customer or employer who comments the work. The system is then remade with the requested modifications. It goes back to the customer again, who again give comments. The system then goes back and forth between developer and customer until it it's completed with all of the requested functionality and the customer is satisfied.

3.7.3 Spiral development model

A spiral development model has a predefined set of phases in the development. It is called a spiral model because when all phases are finished the development starts with phase one again. The spiral can go on for several rounds. For each round, the level of granularity increases, and the focus in the phase can change slightly.

3.7.4 Waterfall

The waterfall model consists of a predefined set of phases the development must go through. The phases come in a strict order, and each phase is only carried out once. The developers must be certain that a phase is completed before moving on to the next phase. This model is considered slightly less efficient than prototyping and spiral development. This is caused by the fact that extra care must be taken to ensure completeness in each phase.

3.7.5 Looping waterfall model

A looping waterfall model is a combination of the waterfall models and either the spiral development model and/or prototyping. The basis for this type of development is still the waterfall model. It has however a slight modification. It allows the developers to redo one or more of the phases at certain stages in the development. It might be to redo one of the phases once, or it might be to loop between two specific phases until a satisfying result has been obtained. Or even some combination of this.

3.7.6 Choice of development model

The student has used the waterfall model several times. He has great experiences with it, and is relatively un-experienced with the other models. It is therefore a natural choice of development model. Considering the level of uncertainty of how this system should be made and what functionality it should provide, it would however be preferable to be able to redo some of the development phases several times. The looping waterfall model is therefore chosen to be used for the rest of the system development.

4 Requirements specification document

The requirements specification document is made to formally describe the requirements of the system to be developed. It explains how the system should work, what it should be able to do as well as possible limitations.

The development of the requirements specification document is done stepwise. First the requirements are collected. Since there is no external customer in this project, the requirements are solely written by the author. The requirements must then be further analyzed and divided into categories. During implementation the requirements must be validated. This is also the case when the system is finished.

In this project the requirements will not be revised during implementation unless there is a specific need for it. Extra care is therefore needed while writing the requirements.

The requirements specification document does not state how the requirements should be implemented. This is done in the system design document. The exceptions are the requirements for the graphical user interface which give directions on the implementation as well.

4.1 The organization of the requirements specification document

The next chapter in the document discuss' the principal structure of the system. It is followed by three chapters containing the actual requirements. Finally a test plan for the system is presented.

Even though this is a rather small system there are lots of requirements. They are therefore divided into three categories to increase the readability of this document. This will also be useful during the system design phase and the development of requirement tests. During the development of a system the requirements must be tested and verified. Possible changes will be easier to carry out when the document is structured in this way.

The three superior categories of requirements in this document are:

Requirements specification document

- Non functional requirements
- Functional requirements
- Requirements to the graphical user interface

The category of non functional requirements is further divided in sub-categories. The requirements' names (their IDs) are given with respect to their category and sub-category. The different categories are described in the respective chapters. The following naming convention is used:

xx – is the numbering of the requirements and start with 01, 02, 03, etc. Non functional requirements begin with NFR, functional requirements with FR and requirement to the graphical user interface with GR. The complete organization is shown in Table 12.

Table 12 Explanation of requirement IDs

Requirement ID	Explanation of requirement ID
NFR-Ixx	Requirements for system implementation
NFR-SWxx	Requirements for software
NFR-Axx	Requirements for system accessibility
NFR-Sxx	Requirements for system security
NFR-UDxx	Requirements for user documentation
NFR-SDxx	Requirements for system documentation
NFR-CDxx	Requirements for code documentation
NFR-TDxx	Requirements for test documentation
FRS1-xx	Functional requirements for the topic map population system
FRS2-xx	Functional requirements for the topic map application system
GR-xx	Requirements for the graphical user interface

Each requirement is presented in a table with the requirements' ID above the table. The table structure is explained in the example below.

Requirement: NFR-Ixx

Description	A short and precise presentation of the requirement.
Explanation	This field is meant to explain the purpose of the requirement

Requirements specification document

	and/or a more detailed description of the requirement.
Importance estimation	An importance estimation is a short discussion of to whom this requirement is important and why.
Importance	In this field a graduation of the requirements importance is presented. Three categories are used: High, medium and low. These are meant to give directions as to what should be left out if there is not enough time to implement all the requirements. In principle all requirements should be implemented. There is however a strict time limit on the system development, and some requirements might be omitted. An explanation of the graduation is given in the next table.

Table 13 Explanation of importance estimation

Importance	Description
High	The requirements of this importance must be implemented for the system to function properly.
Medium	The requirements of this importance are important as well, but are not necessary for the systems' existence.
Low	Requirements of this importance will be the ones who first will be omitted if the time is running out.

4.2 Non functional requirements

The non functional requirements are requirements that do not concern the specific functionality of the system. These requirements should be as measurable and concrete as possible, so their fulfillment can be verified after implementation. The non functional requirements are divided into the following sub-categories:

- Implementation
- Software
- Accessibility
- Security
- User documentation
- System documentation
- Code documentation
- Test documentation

4.2.1 Implementation

These requirements concern how the implementation should be done from a technical point of view.

Requirement: NFR-IO1

Description	The system should be made open source.
Explanation	A system can not be made open source unless all its components are. This includes programming tools, standards, APIs and libraries during implementation.
Importance estimation	Unless this requirement is followed, new developers will not be able to extend the system by their own desire. It is therefore an important requirement. It is however not required by the project group or for the system to work.
Importance	Medium

Requirement: NFR-IO2

Requirements specification document

Description	All programming must be done in English.
Explanation	QUIS is a multilingual project, and English is used as common language. This is also preferable since the system is to be made open source.
Importance estimation	This is an important requirement both because it enables the other project members to understand, and because making the system open source is important.
Importance	High

Requirement: NFR-I03

Description	The implementation must be as modular as possible.
Explanation	The system development is done by one person only, and is limited by time. Not all functionality is certain to be implemented. The system must therefore be able to run the components that are finished, which again requires high modularity and few dependencies. It is also easier to extend a modular system.
Importance estimation	Being able to present a functional system within deadline is very important for the author since the system and this paper will be graded.
Importance	High

Requirement: NFR-I04

Description	All written code should be commented and explained.
Explanation	Good comments and explanations will increase the readability of the code, and simplify further development by other users. This is important as the system is to be open source.
Importance estimation	The importance of this requirement follows as a direct consequence of the importance of requirement NFR-I01 “The system must be made open source”.
Importance	Medium

4.2.2 Software

This requirement concerns software that is needed to be able to use the system.

Requirement: NFR-S01

Description	The system must be accessible by all web browsers supporting JSP.
Explanation	The system will be available from www.tisip.no/quis (later requirement). It is meant to be used by the QUIS projects members and anyone else interested. It must therefore not require any non common programs or plug-ins to be used.
Importance estimation	This is important because it enables anyone with Internet access to test the system.
Importance	High

4.2.3 Accessibility

These requirements concern where to locate the system and the systems response time.

Requirement: NFR-A01

Description	The system must be accessible via the www.tisip.no/quis web page.
Explanation	www.tisip.no/quis is the projects web page. The system must therefore be accessible from that location.
Importance estimation	This is important for giving as many interested people as possible access to the system.
Importance	High

Requirement: NFR-A02

Description	The maximum response time of the system must under normal circumstances be less than three seconds.
Explanation	Users will easily grow tired of a system with longer response time than this.

Requirements specification document

Importance estimation	A system to slow for anyone to use is a bad system, no matter how good its functionality is. This is therefore an important requirement.
Importance	High

Requirement: NFR-A03

Description	When the response time exceeds three seconds the user should receive a message.
Explanation	Such a message can prevent users from not using the system again, if the system is slow due to for instance technical problems.
Importance estimation	This is important, but not a necessity for the system to work.
Importance	Medium

4.2.4 Security

This requirement concerns the systems' security.

Requirement: NFR-SW01

Description	Access to the "Partners' arena" must require a username and a password.
Explanation	The information in the "Partners' arena" is only meant for the members of the QUIS project members. It must therefore be protected.
Importance estimation	The information in the "Partners' arena" requires a username and a password on the projects web pages today. This new system must require the same. It is therefore an important requirement.
Importance	High

4.2.5 User documentation

Good user documentation is always desirable in a system. Requirements for the user documentation are presented in this sub-chapter.

Requirement: NFR-UD01

Description	The user documentation must be written in English.
Explanation	Since the system is open source and can be modified by anyone interested the documentation should be in English.
Importance estimation	Since the common language in the QUIS project is English this is an important requirement.
Importance	High

Requirement: NFR-UD02

Description	The user documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.
Explanation	It should be easy to read the user documentation, in order to enable users that do not have English as their first language to understand it. The language should therefore not be too technical.
Importance estimation	This is important so that users will do the effort of reading it.
Importance	Medium

4.2.6 System documentation

System documentation is important for those who wish to install and extend the system. Requirements for the system documentation are presented here.

Requirement: NFR-SD01

Description	The installation documentation should explain how to install the system.
Explanation	To encourage new developers to do further work on the system

Requirements specification document

	there should be an explanation of how to install the existing one.
Importance estimation	The importance of this requirement falls in the same class as the requirement of the system being open source.
Importance	Medium

Requirement: NFR-SD02

Description	The system documentation must be written in English.
Explanation	Since the system is open source and can be modified by anyone interested the documentation should be in English.
Importance estimation	Since the common language in the QUIS project is English this is an important requirement.
Importance	High

Requirement: NFR-SD03

Description	The system documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.
Explanation	It should be easy to read the system documentation, in order to enable users that do not have English as their first language to understand it.
Importance estimation	This is important so that users will do the effort of reading it.
Importance	Medium

4.2.7 Code documentation

Since the system will be made open source, good code documentation is needed and its requirements follows here.

Requirement: NFR-CD01

Description	The code documentation must be written in English.
Explanation	Since the system is open source and can be modified by anyone interested the documentation should be in English.

Importance estimation	Since the common language in the QUIS project is English this is an important requirement.
Importance	High

Requirement: NFR-CD02

Description	All code should be documented with Javadoc ²⁰ .
Explanation	To make it easy to navigate through the code documentation and to understand it, it is important to use a well known standard.
Importance estimation	This is important both for the system developer and for people working on the system later.
Importance	Medium.

4.2.8 Test documentation

The test documentation is mainly important for the system developer. The requirements are presented in this sub-chapter.

Requirement: NFR-TD01

Description	The test documentation must for each test explain which requirements the test is verifying.
Explanation	It is to be given an explanation for each test. This makes it easier to verify that the test itself is necessary and so that one can control that all requirements are being tested.
Importance estimation	This is important for the developer to verify that the system actually fulfills the requirements given to it.
Importance	High

Requirement: NFR-TD02

Description	Each variant of testing should be explained, and the use of the variant should be reasoned for.
--------------------	---

²⁰ Javadoc is a tool that produces documentation about the code, based on the code itself and the comments within the Java files.

Explanation	It is important to understand why a specific kind of test is being used. This leads to better understanding, execution, control and interpretation of the test and its results.
Importance estimation	This requirement is important for the system developer.
Importance	Medium

4.3 Functional requirements

As the name implies these are requirements to the systems functionality. The requirements must be written in a way so they make a basis for the system design.

It is actually to separate systems who will be made. The first for creating and populating a topic map, and the second for presenting the topic maps content to the end user. The requirements for these systems will be presented to two separate sub-chapters.

4.3.1 Functional requirements for the topic map creator system (system1)

There requirements' IDs all begin with FRS1, meaning *functional requirements for system one*.

Requirement: FRS1-01

Description	It must be possible to add topics to the topic map.
Explanation	A topic map is nothing without topics.
Importance estimation	This is of the highest importance.
Importance	High

Requirement: FRS1-02

Description	It must be possible to type the topics, associations and occurrences in the topic map.
Explanation	Typing of these topic map elements add navigational structure as well as semantic value to the topic map.

Importance estimation	Typing is of the utter most importance.
Importance	High

Requirement: FRS1-03

Description	It must be possible to add associations with members and member roles to the topic map.
Explanation	Associations are one of the three major elements in topic maps. A topic map is nothing without associations.
Importance estimation	This is of the highest importance.
Importance	High

Requirement: FRS1-04

Description	It must be possible to add occurrences to the topic map.
Explanation	Occurrences are one of the three major elements in topic maps. A topic map is nothing without occurrences.
Importance estimation	This is of the highest importance.
Importance	High

Requirement: FRS1-05

Description	It should be possible to add scopes to the topic map elements.
Explanation	Scopes are valuable both when displaying topic map content and when merging topic maps.
Importance estimation	This requirement gives extra functionality to the topic map, but it not necessary for having a useable topic map.
Importance	Medium

4.3.2 Functional requirements for the topic map application system (system2)

There requirements' IDs all begin with FRS2, meaning *functional requirements for system 2*.

Requirement: FRS2-01

Description	The system must give the user access to all topics.
--------------------	---

Requirements specification document

Explanation	Seeing all topics is both valuable when exploring a topic map, as well as for un-experienced users to understand topic map concepts.
Importance estimation	Usability and understandability are important.
Importance	High

Requirement: FRS2-02

Description	The system must give the user access to all topic types.
Explanation	A topic type is a good starting point when exploring a topic map.
Importance estimation	This will increase the usability of the topic map application and is therefore important.
Importance	High

Requirement: FRS2-03

Description	The system must give the user access to all associations.
Explanation	Seeing all associations is both valuable when exploring a topic map, as well as for un-experienced users to understand topic map concepts.
Importance estimation	These are important aspects.
Importance	High

Requirement: FRS2-04

Description	The system must give the user access to all association types.
Explanation	An association type is a good starting point when exploring a topic map.
Importance estimation	This will increase the usability of the topic map application and is therefore important.
Importance	High

Requirement: FRS2-05

Description	The system must give the user access to single topics.
Explanation	Single topics are where most of the actual information about the subjects is.

Requirements specification document

Importance estimation	This is most important.
Importance	High

Requirement: FRS2-06

Description	The system must give the user access to single associations.
Explanation	A single association explains the relationship between one or more topics. This information should be accessible both from the association type typing this association, as well as from the topics that are members of the association.
Importance estimation	This is most important.
Importance	High

Requirement: FRS2-07

Description	The system must give the user access to relevant topic characteristics.
Explanation	While accessing a single topic, the system must present information about: All external and internal occurrences, scoped information, all associations the topic is a member of, the topic that types this topic and the topics that are of this topics' type.
Importance estimation	This is important for the users looking for information about the subjects.
Importance	High

Requirement: FRS2-08

Description	It should be possible to search the topic map.
Explanation	The system should give the user a possibility to search for topics, occurrences and applications via a free text based search engine.
Importance estimation	This increases the usability of the topic map. It is however possible to have a topic map application without search functionality.
Importance	Medium

4.4 Requirements to the graphical user interface

In this chapter the requirements to the graphical user interface are presented. The graphical user interface is the users' way of accessing the system.

Requirement: GR-01

Description	The user should interact with the system via links and search bars.
Explanation	This is the common way to navigate in web structure and it should also be the case in the topic map application. It presents an easy way to requested information.
Importance estimation	This is important. The presence of a search bar is however of medium importance.
Importance	Medium

Requirement: GR-02

Description	Input to the system must be done with the mouse, the keyboard and via file input.
Explanation	These are input units available for most users.
Importance estimation	It is important that the input units are units available for most users.
Importance	High

Requirement: GR-03

Description	Response from the system must be visual on screen.
Explanation	A graphical user interface is desirable. It is important that the user receives information interactively via the screen.
Importance estimation	This requirement is most important.
Importance	High

Requirement: GR-04

Requirements specification document

Description	The system should not present any “surprising” elements.
Explanation	Error messages without explanation or un-logical events should not appear.
Importance estimation	Such messages or events would displease the user and possibly make him stop using the program.
Importance	Low

Requirement: GR-05

Description	The system should have a maximum of seven new elements on each screen level.
Explanation	This will reduce the cognitive effort needed from the user.
Importance estimation	This is important when a new user is getting familiar with the system
Importance	Low

4.5 Summary of the requirements

This chapter presents a summary of all the systems requirements. This makes it easier to locate the requirements for later purposes and potential evaluation of which requirements can be left out. The requirements are sorted after importance and belonging. The requirements are presented in Table 14.

Table 14 Summary of the system requirements

Requirement ID	Description	Importance
NFR-I02	All programming must be done in English.	High
NFR-I03	The implementation must be as modular as possible.	High
NFR-SW01	The system must be accessible by all web browsers supporting JSP.	High
NFR-A01	The system must be accessible via the www.tisip.no/quis web page.	High
NFR-A02	The maximum response time of the system must under normal circumstances be less than three seconds.	High

Requirements specification document

NFR-S01	Access to the “Partners’ arena” must require a username and a password.	High
NFR-UD01	The user documentation must be written in English.	High
NFR-SD02	The system documentation must be written in English.	High
NFR-CD01	The code documentation must be written in English.	High
NFR-TD01	The test documentation must for each test explain which requirements the test is verifying.	High
FRS1-01	It must be possible to add topics to the topic map.	High
FRS1-02	It must be possible to type the topics, associations and occurrences in the topic map.	High
FRS1-03	It must be possible to add associations with members and member roles to the topic map.	High
FRS1-04	It must be possible to add occurrences to the topic map.	High
FRS2-01	The system must give the user access to all topics.	High
FRS2-02	The system must give the user access to all topic types.	High
FRS2-03	The system must give the user access to all associations.	High
FRS2-04	The system must give the user access to all association types.	High
FRS2-05	The system must give the user access to single topics.	High
FRS2-06	The system must give the user access to single associations.	High
FRS2-07	The system must give the user access to relevant topic characteristics.	High
GR-02	Input to the system must be done with the mouse, the keyboard and via file input.	High
GR-03	Response from the system must be visual on screen.	High
NFR-I01	The system should be made open source.	Medium
NFR-I04	All written code should be commented and explained.	Medium
NFR-A03	When the response time exceeds three seconds the user should receive a message.	Medium
NFR-UD02	The user documentation should be written in a way that	Medium

Requirements specification document

	it can be read and understood by a person having English as a second language within 30 minutes.	
NFR-SD01	The installation documentation should explain how to install the system.	Medium
NFR-SD03	The system documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.	Medium
NFR-CD02	All code should be documented with Javadoc.	Medium
NFR-TD02	Each variant of testing should be explained, and the use of the variant should be reasoned for.	Medium
FRS1-05	It should be possible to add scopes to the topic map elements.	Medium
FRS2-08	It should be possible to search the topic map.	Medium
GR-01	The user should interact with the system via links and search bars.	Medium
GR-04	The system should not present any “surprising” elements.	Low
GR-05	The system should have a maximum of seven new elements on each screen level.	Low

4.6 Superior test plan

This sub-chapter contains the superior test plan of the system. The test plan does not contain the actual tests, but a survey of which tests that will be carried out, and how they will be carried out and evaluated. A separate test document follows later in chapter 7.

During the development of a software system, continuous testing is necessary to eliminate code errors and to validate the systems' integrity. The elimination of code errors is called verification, which means to ensure that “the system is made correctly”. Validation means to ensure that the system does what it is intended to do; to ensure that “the correct system is made”. To ensure consistent testing a test plan is needed in advance. The main elements in this test plan follows in the next sub-chapters.

4.6.1 Tests to be carried out

The tests will be carried out by the system developer. Since this system is a part of a master thesis and not made on request from a customer, no customer testing is needed. The objective of these tests is to ensure that the programming code is without errors, and that the interactions between the different system modules work correctly. The tests are performed in a matter that the developer is not bothered with errors from an earlier test stage. The tests are divided into three levels; unit tests, module tests and system and integration test, in the following way:

- “Unit tests” test the smallest units of programming code. This includes classes, objects, forms, graphics, etc.
- The “module tests” ensure that the different modules the system is made up of work as single units, and that the interaction between them works. The design of these tests is done in the design phase of the system.
- The “system and integration test” is a complete test of the system as a whole. Testing for accordance to the system requirements is carried out here.

4.6.2 Check lists

Check lists are simple lists, like sequential test recipes, used during unit and module testing.

4.6.3 Test specifications

A test specification is a more detailed description of how to carry out a test. Among other things it contains information about:

- Data input to the test
- Expected test results
- Test duration
- Error reporting

A more detailed description of the test specification follows in the separate test document.

4.6.4 Black box and white box testing

Black box and white box testing are to common test methods, which will be used in this project. More information about these methods follows in the separate test document.

4.6.5 Test evaluation

A more thorough description of the criteria of test evaluation follows in the separate test document.

5 System design document

The objective for the system design document is to present how the system is meant to be designed and constructed. This is used as a structural survey of how to do the actual implementation. The design is based on the discussion of possible solutions and technologies in the pre study, as well as the requirements in the requirements specification document.

First a superior system description is presented. After that a more detailed description of the different system modules. The different modules to be described are as follows:

- **Topic map constructor system**

This chapter presents how the topic map constructor system will be constructed. First an introduction is given. Then the different functional aspects of the system are presented, based on the requirements in the requirements specification document. Finally the graphical user interface and its structure are presented.

- **Topic map application java module**

This chapter presents the java module of the topic map application system. This includes the different classes, the functions they offer and the interaction between the classes.

- **Topic map application server and jsp module**

This chapter presents the construction and design of the topic map application server and jsp module. This part of the system will mainly receive a request from the end user, which is passed on to the topic map application java module. The response from the java module is then passed on back to the server and jsp module, and the end user receives the answer of his request.

5.1 Superior system description

This chapter presents a superior system description in order to make it easier to achieve an overview of the system as a whole. The total system will first be described, before a brief view of the system modules is given. Finally the catalogue and package structure used for implementation is presented.

5.1.1 System description

Figure 8 shows how the system is physically divided with a server and a client computer, and which modules are physically located where. The figure further shows the main division of modules and the communication between them. The part of the figure in the red square is the topic map constructor system. This is used by the topic map administrator and outputs a topic map used in the topic map application system. The topic map application system is shown in the blue square in the figure.

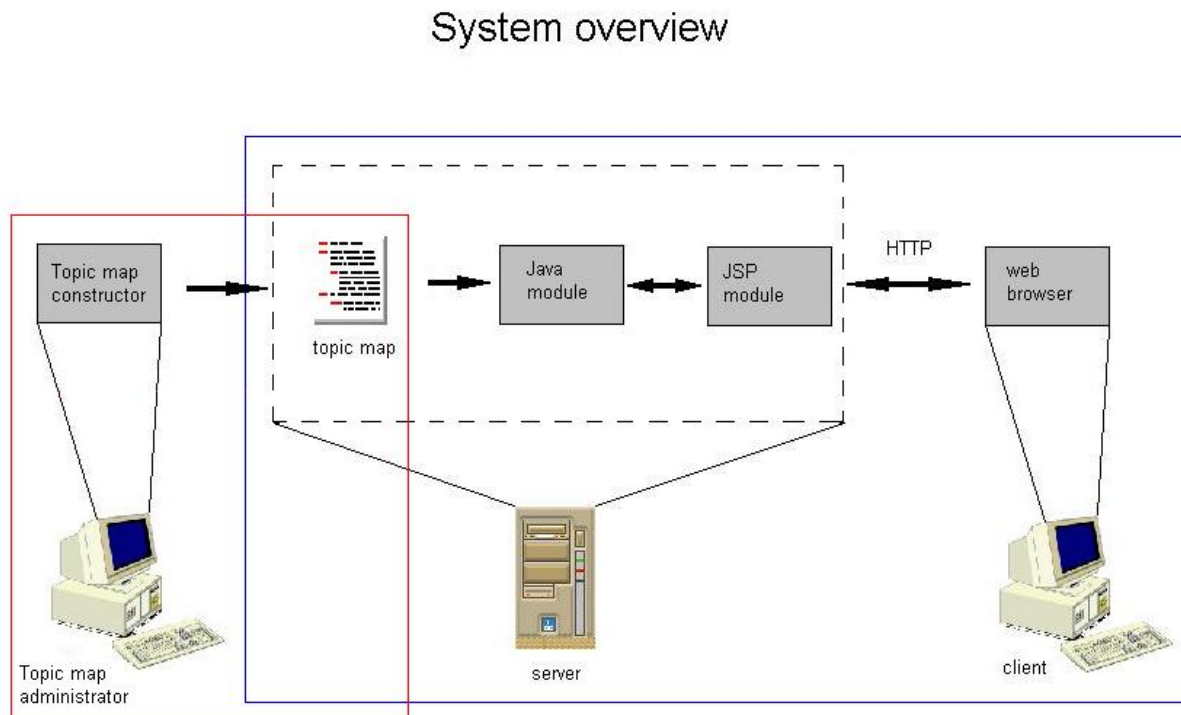


Figure 8 System overview

Figure 9 shows a survey of the superior system calls, between the different modules. First the client chooses the topic map element he wishes to see. This results in a “request” with relevant parameters to the server module. The server then sends a “request” to the java

module, which again collects the actual data from the topic map. The java module sends the data to the server module, and the server module finally generates a “response” to the client.

Superior sequential diagram

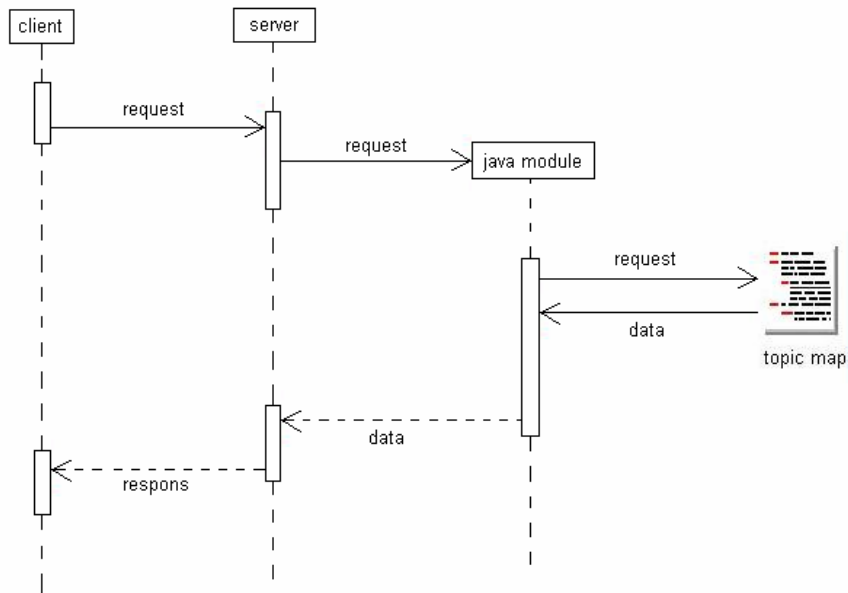


Figure 9 Superior sequential diagram

5.1.2 System modules

The following chapters present a short description of the system modules.

The client module

This is mentioned even though a separate client module is not developed in this system. The end user or client uses a web browser to examine the topic map. The client module will be given no further explanation.

The server/JSP module

This module manages all requests from the client module. It is build from JSP and uses Tomcat as web application server. The configuration of this module is described in chapter 5.2.

The java module

This module communicates with the server module and collects relevant data from the topic map. A further explanation is given in chapter 5.3.

The topic map

It is a bit exaggerated to call the topic map a module. It is however a separate part of the system, and is being used in both the topic map constructor and the topic map application. The topic map is thoroughly discussed in chapter 2.

The topic map constructor module

The topic map constructor module either build a topic map from scratch or updates a topic map by inserting new elements or removing existing ones. A further explanation is given in chapter 5.4.

5.2 The server/JSP module

The objective for this chapter is to describe the server/JSP module which among other things manages requests from the client. We begin with examining the web application server which is the container for the server module. Then the JSP part of the module is described.

5.2.1 Web application server

The main responsibility for the web application server is to receive requests from the client, process these, and send a reply to the client in a known format. In our case this format is HTML since the client is a web browser.

In this system Tomcat is used as application server. Tomcat is further described chapter 3.6.4. Figure 10 shows a block diagram for the application server and its communication with the client. Tomcat is a Java application and uses the Java API.

Block diagram for application server to client communication

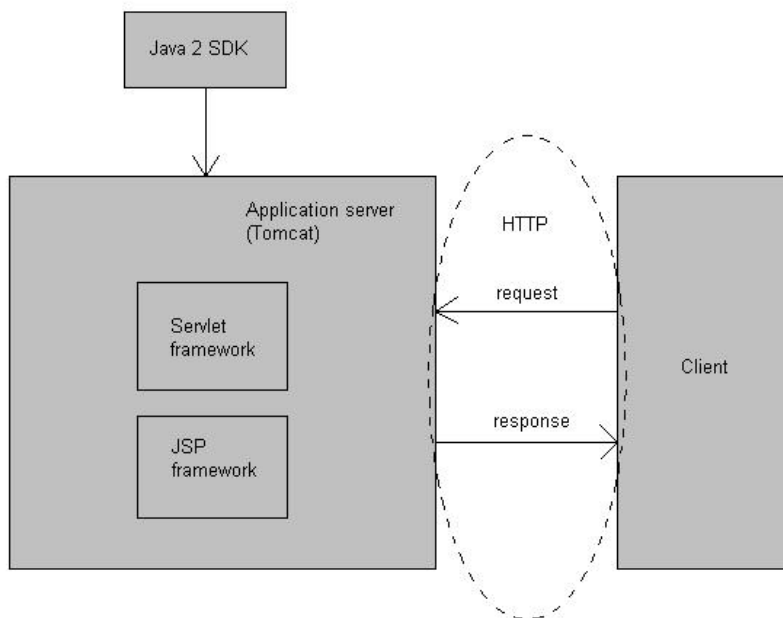


Figure 10 Server-Client block diagram

The servlet framework has responsibility for running Java Servlets. A servlet runs as a Java application and has direct access to the response object on the server. A more closely description of servlets is given in the pre study appendix 3.6.3. JSP programs are run using the JSP framework. JSP is also further described in the pre study.

Figure 11 shows a superior sequential diagram for the server-client communication.

Sequential diagram for client-server communication

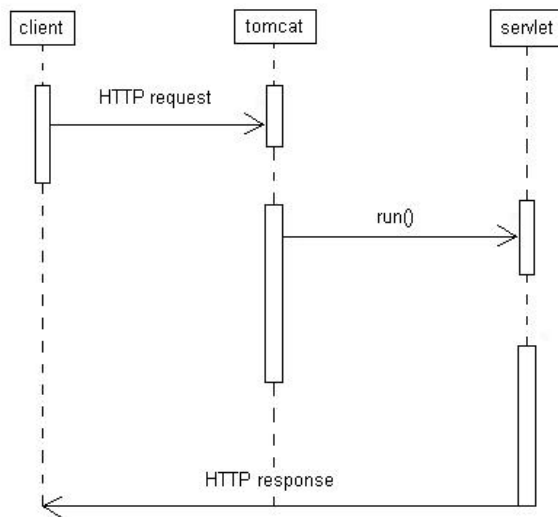


Figure 11 Sequential diagram for client-server communication

When the application server receives a request, the servlet made to manage this is run. The results from the servlet is written to a response object and sent back to the client.

5.2.2 The JSP module

Java Server Pages are HTML pages where Java code can be written inside the HTML code. The entire Java API can be used, as well as all other Java classes and libraries. By using JSP logic and layout are mixed on the same pages. This is not advantageous, but due to the size of the system being made, it will not be a problem. Further arguments for using JSP are presented in the pre study. Figure 12 shows a page diagram over the JSP pages used in the system.

JSP page diagram

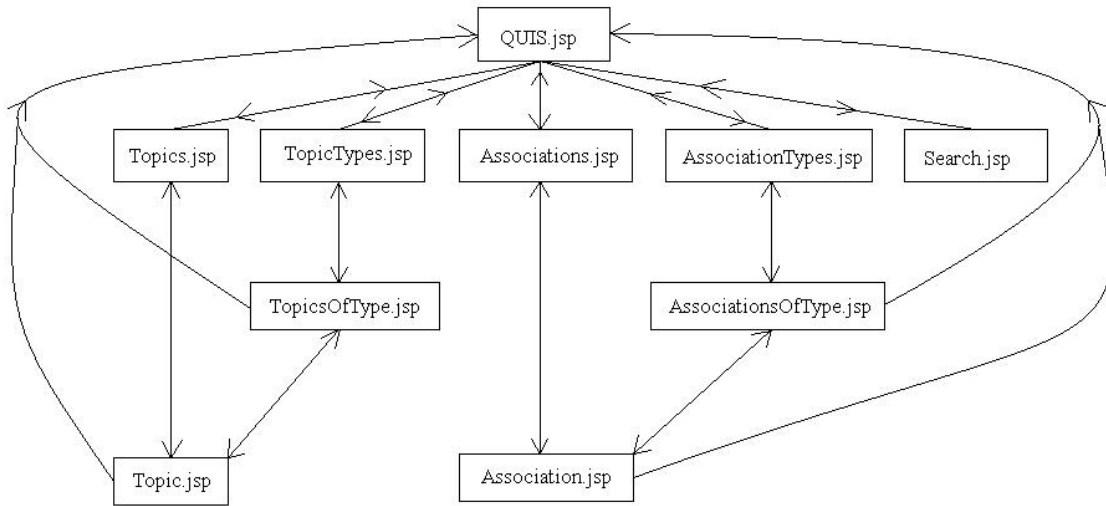


Figure 12 JSP page diagram

Page description

A short description of the different JSP pages will be given in this sub-chapter.

- **QUIIS.jsp**

This is the home page of the system. It should give a brief introduction to the system and possibly contain links to help pages. Help pages and other pages with information not contained in the topic map file are not included in the figure. The home page contains links to five different pages with topic map information: *Topics.jsp*, *TopicTypes.jsp*, *Associations.jsp*, *AssociationTypes.jsp* and *Search.jsp*. Every page in the system has a link that leads back to the QUIIS.jsp home page. This page creates an instance of the *QuisTopicMap* class and the *TopicExtractorQuis*, *AssociationExtractorQuis* and *SearchTopic* classes. These objects are saved in a Java Bean for further use by the other JSP pages.

- **Topics.jsp**

This page shows a list of all the topics in the topic map. This is useful if the user does not know the type of the topic of interest. It is also practical for exploring the topic map. From this page the user can either go back to the home page or choose to further investigate one of the topics. The page calls the *getTopics()* method in the *TopicExtractorQuis* object.

- **TopicTypes.jsp**

This page shows a list of the typing topics in the topic map; the topic types. From here the user can either choose to get a list of the different topics of a specific topic type, or go back to the home page. The page calls the *getTopicTypes()* method in the *TopicExtractorQuis* object.

- **TopicsOfType.jsp**

This page shows a list of the different topics of a specific type, chosen in the *TopicTypes.jsp* page. From here the user can either choose to go back to the *TopicTypes* or the home page, or to further investigate one of the topics shown. The page calls the *getTopicsOfType()* method in the *TopicExtractorQuis* object.

- **Topic.jsp**

This page shows information about a specific topic either chosen from the *Topics* or the *TopicsOfType* page. The information shown is occurrences, associations, alternative names and scoped information. From this page the user can go back to the home page, the *Topics* page or the *TopicsOfType* page. The page calls the *getTopicElements()* method in the *TopicExtractorQuis* object.

- **Associations.jsp**

This page shows a list of all the associations in the topic map. From this page the user can either go back to the home page or choose to further investigate one of the associations. The page calls the *getAssociations()* method in the *AssociationExtractorQuis* object.

- **AssociationTypes.jsp**

This page shows a list of the typing associations in the topic map; the association types. From here the user can either choose to get a list of the different associations of a specific association type, or go back to the home page. The page calls the *getAssociationTypes()* method in the *AssociationExtractorQuis* object.

- **AssociationsOfType.jsp**

This page shows a list of the different associations of a specific type, chosen in the Associations page. From here the user can either choose to go back to the Associations or the home page, or to further investigate one of the associations shown. The page calls the *getAssociationsOfType()* method in the *AssociationExtractorQuis* object.

- **Association.jsp**

This page shows information about a specific topic either chosen from the Associations or the AssociationsOfType page. The information shown is the associations' members and their roles. From this page the user can go back to the home page, the Associations page or the AssociationsOfType page. The page calls the *getAssociationElements()* method in the *AssociationExtractorQuis* object.

- **Search.jsp**

This page offers search functionality to the user. Both topics and associations can be searched for. The page calls the *searchMap()* method in the *TopicSearcher* object.

5.3 The Java module

To get an overview over the different parts of the Java module a class diagram is shown in Figure 13. The class diagram is not complete when it comes to the different class methods. Methods will be created as they are needed. The different classes in the Java module are:

- **QuisTopicMap**

This class loads the topic map file into the system. It contains methods for creating objects of the *TopicExtractorQuis* and *AssociationExtractorQuis* classes. Objects of these classes should only be created here.

- **TopicExtractorQuis**

This class contains methods for retrieving topic map information like the topics and the topic types in the topic map. It also contains a method for retrieving topic elements from a specific topic. These elements being associations, internal and external occurrences, alternative base names and scoped information. The actual extraction of information from the topic map is not done in these methods, but in the methods of the *ExtractTopic* class. The reason for this is to keep the *TopicExtractorQuis* class as simple as possible, making a basis for an interface.

- **ExtractTopic**

This class contains the actual methods for extracting information from the topic map. Objects of this class are created from the *TopicExtractorQuis* class.

- **AssociationExtractorQuis**

This class contains methods for retrieving topic map information like the associations and the association types in the topic map. It also contains a method for retrieving association elements from a specific association. These elements are the members of the association and their roles, alternative association names and scoped information. The actual extraction of information from the topic map is done in the methods of the *ExtractAssociation* class.

- **ExtractAssociation**

This class contains the actual methods for extracting information about the associations in the topic map. Objects of this class are created from the *AssociationExtractorQuis* class.

- **TopicSearcher**

This class contains methods for making indexed searches in the topic map.

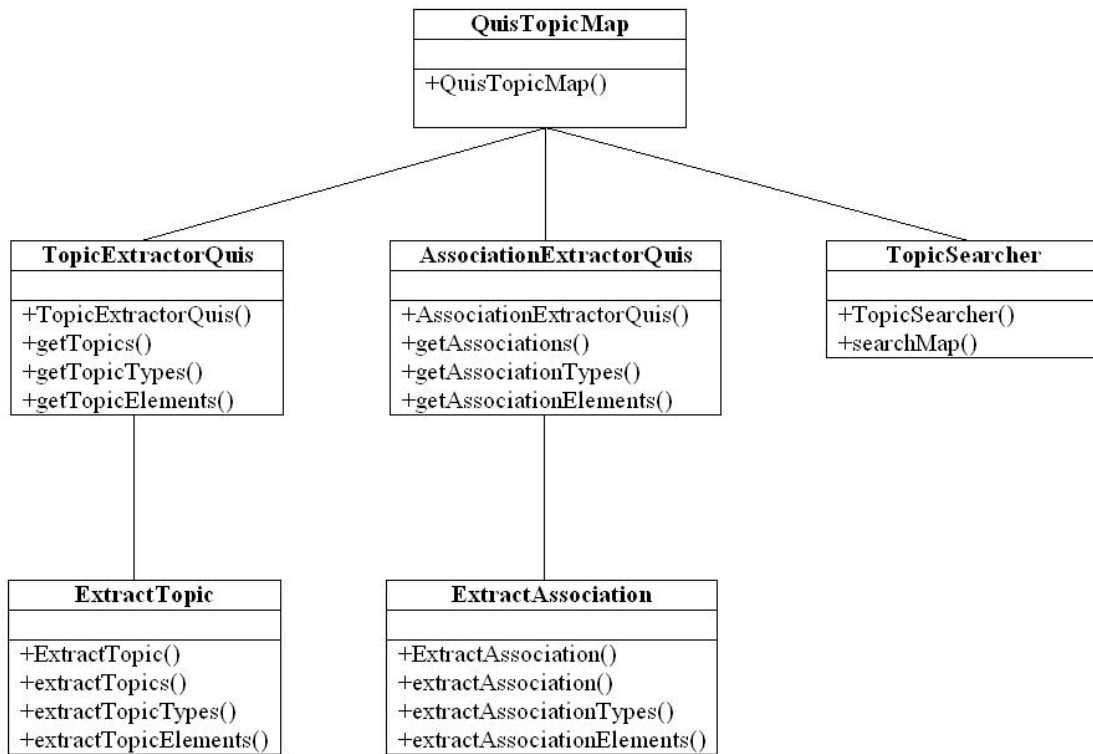


Figure 13 Java module class diagram

5.4 The topic map constructor module

This chapter describes the topic map constructor module or system. This system is solely written in java. The system can perform two similar tasks. Either to import an existing topic map and add and/or remove elements, or to build a new topic map from scratch. The topic map build in this system could later be used in the topic map application system. Figure 14 shows a class diagram of the system.

Topic map constructor system class diagram

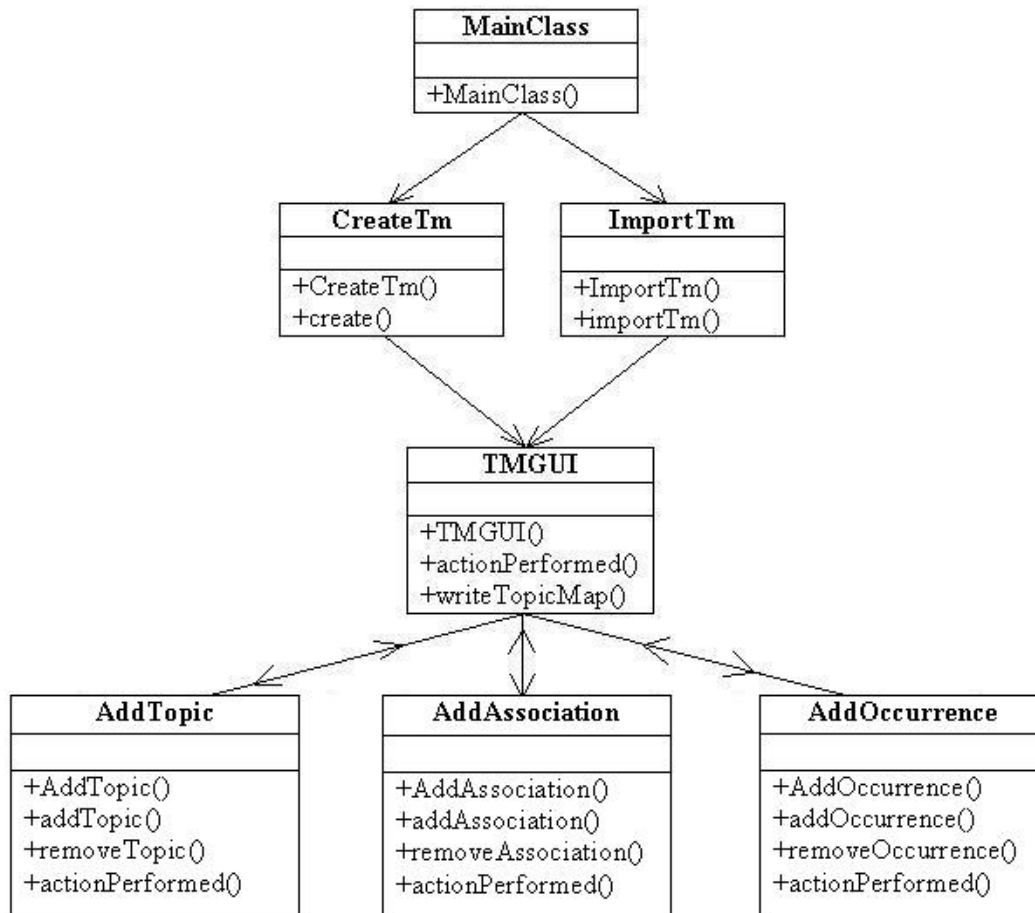


Figure 14 Topic map constructor system class diagram

The diagram is not complete when it comes to class methods. Other methods will be created as they are needed in the implementation phase. The different classes are:

- **MainClass**

The MainClass opens the system. Depending on the input arguments it either creates an object of CreateTm or ImportTm.

- **CreateTm**

CreateTm opens a new file with name according to the user input arguments given to MainClass. It then creates an object of TMGUI and passes the new file as argument.

- **ImportTm**

ImportTm takes an existing topic map file as argument, opens this file, creates a new object of TMGUI and passes the topic map file as argument.

- **TMGUI**

TMGUI is build using the Java Swing library, and presents the main menu to the user. The user can choose to add and/or remove topics, associations and occurrences. According to the users' choice, objects of type AddTopic, AddAssoication or AddOccurrence are created and opened. These classes are also GUI classes created with the Java Swing library. The TMGUI also gives the user the possibility to end the application.

- **AddTopic**

This class is a GUI class, allowing the user to either add or remove topics to/from the topic map. When AddTopic is closed, the application returns to TMGUI.

- **AddAssociation**

This class is a GUI class, allowing the user to either add or remove associations to/from the topic map. When AddAssociation is closed, the application returns to TMGUI.

- **AddOccurrence**

This class is a GUI class, allowing the user to either add or remove occurrences to/from the topic map. When AddOccurrence is closed, the application returns to TMGUI.

Figure 15 below shows a data flow diagram (DFD) of the topic map constructor system.

Data flow diagram for the topic map constructor system

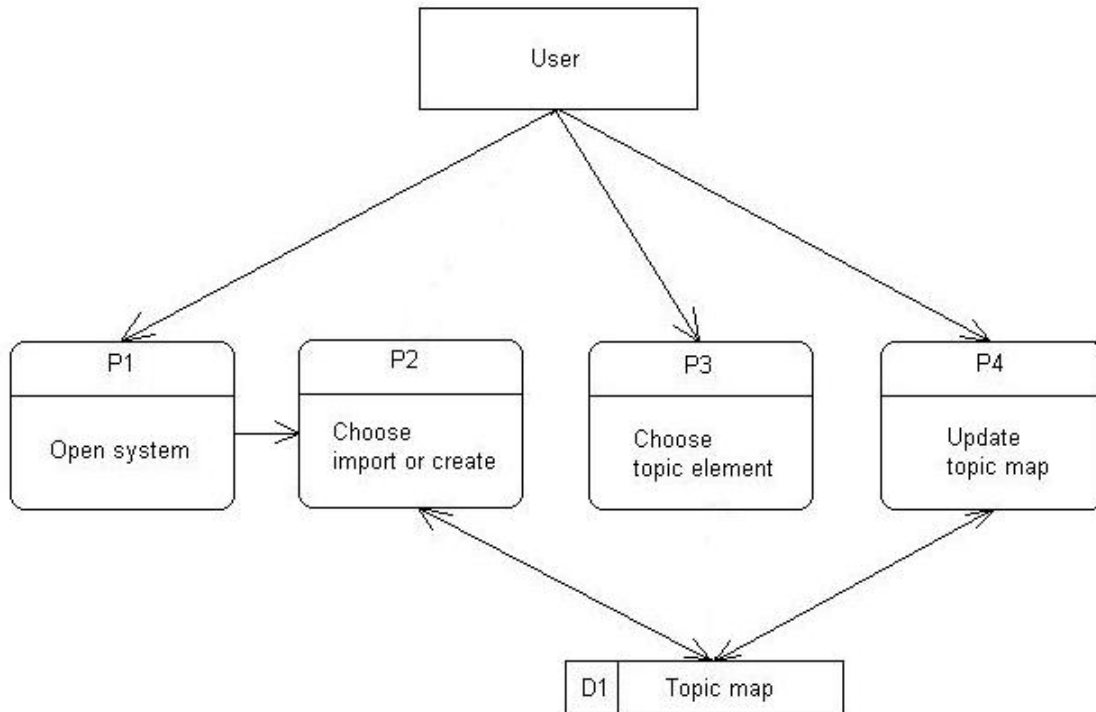


Figure 15 Data flow diagram for the topic map constructor system

- **P1 – Open system**

The user, in this case the topic map administrator, opens the system by running the MainClass. The user input is either an existing topic map file or the reference to a new file to be created. The user also specifies whether the file argument is a new or an existing file.

- **P2 – Choose import or create**

The system examines the input arguments and creates the respective objects. If the input argument is an existing file, the file is loaded from **D1**. If the argument is a reference to a new file, the file is created and stored in **D1**. The user is prompted the main GUI.

- **P3 – Choose topic element**

The user chooses to add and or remove topics, associations or occurrences.

System design document

- **P4 – Update topic map**

The user adds and/or removes topic map elements to/from the topic map. The topic map **D1** is updated with the information.

- **D1 – Topic map**

The topic map xml (XTM) file.

6 Implementation and documentation

The implementation and documentation document gives examples of the programming code as well as the different documentations. The complete programming code is delivered on a separate CD. The programming examples will be presented first. On class from the topic map constructor system, one class from the topic map application system and one of the JSP pages. Thereafter follows an Installation Guide and a User's Guide.

6.1 Programming code examples

This sub-chapter presents three different code examples with description of the code. Some insight and experience with Java²¹, TM4J²² and topic maps in general²³ is necessary to fully understand these examples. The rest of this chapter can be left out for those that are not particularly interested.

6.1.1 Topic map constructor code example

This example shows the AddTopic class which is made with the Swing library. It gives the user a graphical interface for adding new topics to the topic map. The user can add the topic name and if wanted a topic type to the topic. Explanations are given as comments in the code. These comments begin with “//” and are written in green.

```
/**
 * AddTopic.java
 */

// Importing standard Java classes

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.*;
import java.util.*;
import java.lang.reflect.Array;

// Importing TM4J classes
import org.tm4j.topicmap.*;
import org.tm4j.net.*;
import org.tm4j.topicmap.memory.TopicMapProviderFactoryImpl;
import org.tm4j.topicmap.source.*;
import org.tm4j.topicmap.utils.extractors.*;
import org.tm4j.topicmap.utils.TopicMapWalker;
```

²¹ <http://java.sun.com/>

²² <http://tm4j.org/>

²³ <http://www.topicmaps.org/>

Implementation and documentation

```
import org.tm4j.topicmap.utils.XTMWriter;

// Importing other classes
import org.apache.xml.serialize.OutputFormat;
import org.apache.xml.serialize.XMLSerializer;

public class AddTopic implements ActionListener {

    // Class variables
    JFrame addTopicFrame;
    JPanel addTopicPanel;
    JTextField topicNameField;
    JLabel topicNameLabel, topicAddedLabel;
    JButton addTopic, exitButton;
    JComboBox topicType, topicTypeExtracted;
    JLabel topicTypeLabel;

    File outfile;
    FileOutputStream os;
    Boolean topicTypeSelected = false;

    TopicMap tm;
    Topic topicInArray, topicTypeTopic, testTopic;
    BaseName baseName;
    TopicNameExtractor topicNameExtractor = new TopicNameExtractor();
    BaseNamesExtractor baseNamesExtractor = new BaseNamesExtractor();

    Object[] baseNameObjectArray;
    String[] topicNames;
    Object[] objectArray;

    Set topicSet;
    ArrayList topicArrayList;
    Collection topicNameCollection;
    String baseNameString;
    String topicTypeChosen = null;

    int i = 0;
    int j = 0;
    int numberOfTopics, collectionSize;

    // Constructor
    public AddTopic2() {
        System.out.println("AddTopic created");
    }

    // Method called from external objects
    public void topicAdder(TopicMap inMap) {

        tm = inMap; // Copying the incoming topic map to the local topic map

        // Creating OutputStream for printing the results
        try {
            outfile = new File("c:/diplom/egne eksempler/test.xtm");
            os = new FileOutputStream(outfile);
        } catch (java.lang.NullPointerException ex) {
            System.err.println(ex.toString());
        } catch (FileNotFoundException ex) {
            System.err.println(ex.toString());
        }

        // Make sure we have nice window decorations.
        JFrame.setDefaultLookAndFeelDecorated(true);

        // Create and set up the window.
        addTopicFrame = new JFrame("Add Topic");
        // addTopicFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addTopicFrame.setSize(new Dimension(240, 120));
    }
}
```

Implementation and documentation

```
// Create and set up the panel.
addTopicPanel = new JPanel(new GridLayout(4, 2));

// Add the widgets.
addWidgets();

// Set the default button.
addTopicFrame.getRootPane().setDefaultButton(addTopic);

// Add the panel to the window.
addTopicFrame.getContentPane().add(addTopicPanel,
BorderLayout.CENTER);

// Display the window.
addTopicFrame.pack();
addTopicFrame.setVisible(true);
}

/**
 * Create and add the widgets.
 */
public void addWidgets() {
    // Create widgets.

    // Where the user enters the new topic name
    topicNameField = new JTextField(4);
    topicNameLabel = new JLabel("Type topic name: ", SwingConstants.LEFT);
    addTopic = new JButton("Add Topic");

    // The topics that can be chosen as topic type
    topicTypeLabel = new JLabel("Choose topic type: ",
SwingConstants.LEFT);
    topicSet = tm.getTopics();

    exitButton = new JButton("Exit");
    topicAddedLabel = new JLabel("...", SwingConstants.LEFT);

    numberOfTopics = tm.getTopicCount();
    System.out.println("number of topics = " + numberOfTopics);
    topicNames = new String[numberOfTopics];
    objectArray = new Object[numberOfTopics];
    objectArray = topicSet.toArray(objectArray);

    // Loop that collects existing topics and retrieves their basenames
    do { // while(objectArray[i]!=null)
        topicInArray = (Topic) (objectArray[i]);
        System.out.println("i = " + i);
        // Retrieving topic BaseNames
        topicNameCollection = (Collection) baseNamesExtractor
            .fn(topicInArray);

        // Converting Collection to Object[]
        Object[] objectArray2 = topicNameCollection.toArray();

        // Retrieving first BaseName in Object[]
        String utNavn2 = ((BaseName) (objectArray2[0])).getData();

        topicNames[i] = utNavn2;

        // Incrementing counter i, and collecting next topic from
        // objectArray[]
        j++;
    } while (j < numberOfTopics);

    // The actual combobox wher the user choses the topic type
    topicType = new JComboBox(topicNames);
```

Implementation and documentation

```
topicType.addActionListener(this);

// Listen to events from the Convert button.
addTopic.addActionListener(this);
exitButton.addActionListener(this);

// Add the widgets to the container.
addTopicPanel.add(topicNameLabel);
addTopicPanel.add(topicNameField);
addTopicPanel.add(topicTypeLabel);
addTopicPanel.add(topicType);
addTopicPanel.add(addTopic);
addTopicPanel.add(topicAddedLabel);
addTopicPanel.add(exitButton);

topicNameLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));
topicAddedLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));
5));
}

// Method that is activated when the application is used
public void actionPerformed(ActionEvent event) {

    // Collects the topic name entered by the user
    String topicName = (String) (topicNameField.getText());

    try {

        // Closes window, but not application
        if (event.getSource() == exitButton) {
            System.out.println("Closing Add Topic Window");
            addTopicFrame.setVisible(false);
        }

        // Collects the chosen topic type
        else if (event.getSource() == topicType) {
            System.out.println("Linje 187");
            topicTypeExtracted = (JComboBox) event.getSource();
            String name = (String)
topicTypeExtracted.getSelectedItem();
            System.out.println(name);
            topicTypeChosen = name;
            topicTypeSelected = true;
        }

        // Creates a new topic
        else {

            // No topic type is chosen for the new topic
            if (topicTypeSelected == false) {
                // Create the new topic
                testTopic = tm.createTopic(topicName);

                // Add a basename
                baseName = testTopic.createName("basename" +
topicName);

                // Add a value to the new basename
                baseName.setData(topicName);

                // Connect the basename with the topic
                testTopic.addName(baseName);

                // Information to the user
                topicAddedLabel.setText("Topic " + topicName + "
added.");
            }
        }
    }
}
```

Implementation and documentation

```
next                                     // Updates the list of possible topic types for the

                                        // topic to be added
topicType.addItem(topicName);
topicType.repaint();
}

// Topic type is chosen
if (topicTypeSelected == true) {

    // Collect the chosen topic type
    topicTypeTopic = tm.getTopicByID(topicTypeChosen);

    // Create the new topic
    testTopic = tm.createTopic(topicName);

    // Add the topic type to the topic
    testTopic.addType(topicTypeTopic);

    // Create basename and add it to the topic
    baseName = testTopic.createName("basename" +

topicName);

    baseName.setData(topicName);
    testTopic.addName(baseName);

    // Information to the user
    topicAddedLabel.setText("Topic " + topicName + "

added.");

                                        // Updates the list of possible topic types for the
next

                                        // topic to be added
topicType.addItem(topicName);
topicType.repaint();
}

}

// Handling errors
catch (Exception ex) {
    System.err.println(ex.toString());
}

}

// Method that writes the topic map to file
public void writeTopicMap(TopicMap tm, OutputStream os) {
    TopicMapWalker walker = new TopicMapWalker();
    XTMWriter writer = new XTMWriter();
    OutputFormat of = new OutputFormat();
    of.setEncoding("UTF-8");
    of.setIndenting(true);
    of.setIndent(2);
    XMLSerializer serializer = new XMLSerializer(os, of);
    walker.setHandler(writer);
    writer.setContentHandler(serializer);
    try {
        walker.walk(tm);
    } catch (TopicMapProcessingException ex) {
        throw new RuntimeException("Unable to write topic map."
            + ex.toString());
    }
}
}
```

6.1.2 Topic map application system code example

The class in this example is the class used by the JSP pages to get all topics and topic types.

This is done in two steps:

- First a call to the method “collectTopics()” is made. This method uses another class for the actual collection of topics and topic types. These are then saved in local variables: “String[] topicsReturnArray;” and “String[] topicTypesReturnArray;”.
- These arrays are then available to the JSP pages through the methods “getTopics()” and “getTopicTypes()”.

```
// Package declaration
package tm4jweb;

// Import classes from Java API
import java.io.*;
import java.net.*;
import java.util.*;
import java.lang.*;

// Import TM4J classes
import org.tm4j.topicmap.*;
import org.tm4j.net.*;
import org.tm4j.topicmap.memory.TopicMapProviderFactoryImpl;
import org.tm4j.topicmap.source.*;
import org.tm4j.topicmap.utils.extractors.*;
import org.tm4j.topicmap.utils.TopicMapWalker;
import org.tm4j.topicmap.utils.XTMWriter;

// Import other classes
import org.apache.xml.serialize.OutputFormat;
import org.apache.xml.serialize.XMLSerializer;

public class TopicExtractorQuis{

    // Class variables

    public BaseNamesExtractor baseNamesExtractor = new BaseNamesExtractor();
    public TopicMap tm = null;

    Collection topicNameCollection;
    OccurrencesExtractor occurrencesExtractor = new OccurrencesExtractor();

    File outfile, outfile2;
    File sourceFile;
    FileOutputStream os;

    SerializedTopicMapSource serializedSource;

    String src;
    String[] topicsReturnArray;
    String[] topicTypesReturnArray;

    //Creating OutputStream for printing the results
    public void openStream(){

    try{
        outfile = new File("c:/diplom/egne eksempler/test2.xtm");
```

Implementation and documentation

```
outfile2 = new File("c:/diplom/egne eksempler/test2.txt");
os = new FileOutputStream(outfile);
}
catch(java.lang.NullPointerException ex){
    System.err.println(ex.toString());
}
catch(FileNotFoundException ex){
    System.err.println(ex.toString());
}
}

// Method that takes the address of a topic map as a parameter
// and loads the topic map into memory
public void importer(String source){
    try{
        src = source;
        sourceFile = new File(src);
        serializedSource = new SerializedTopicMapSource(sourceFile);
        TopicMapProviderFactory tmpf =
TopicMapProviderFactory.newInstance();
        TopicMapProvider provider =
tmpf.newTopicMapProvider(System.getProperties());
        tm = provider.addTopicMap(serializedSource);
    }

    catch(TopicMapProviderFactoryConfigurationError ex){
        System.err.println("Error in specification of provider factory:
" + ex.toString());
    }

    catch(TopicMapProviderException ex){
        System.err.println("Error creating back-end connection: " +
ex.toString());
    }

    catch(FileNotFoundException ex){
        System.err.println("File not found");
    }

    catch(MalformedURLException ex){
        System.err.println("MalformedURLException");
    }
}
return;
}

// Method that writes the topic map to file
public void writeTopicMap(TopicMap tm, OutputStream os)
{
    TopicMapWalker walker = new TopicMapWalker();
    XTMWriter writer = new XTMWriter();
    OutputFormat of = new OutputFormat();
    of.setEncoding("UTF-8");
    of.setIndenting(true);
    of.setIndent(2);
    XMLSerializer serializer = new XMLSerializer(os, of);
    walker.setHandler(writer);
    writer.setContentHandler(serializer);
    try
    {
        walker.walk(tm);
    }
    catch(TopicMapProcessingException ex)
    {
        throw new RuntimeException("Unable to write topic map." +
ex.toString());
    }
}
}
```


Implementation and documentation

```
// Method that collects the names of all topics and saves these
// in a local String[]. It does the same with all topic types.
public void collectTopics(){

    // The actual collection of the topic names is done in another
    // class called ExtractTopic.

    ExtractTopic et = new ExtractTopic();
    String[] topicArr = et.extractTopic(tm);
    String[] topicTypes = et.getTopicTypes();

    topicsReturnArray = topicArr;
    topicTypesReturnArray = topicTypes;

}

// Method that returns all topics
public String[] getTopics(){
    return topicsReturnArray;
}

// Method that returns all topic types
public String[] getTopicTypes(){
    return topicTypesReturnArray;
}
}
```

6.1.3 JSP page code example

The JSP example is a file called topics.jsp. It is a page that displays an array of all the topics in the topic map to the user. Each topic comes with a link that will take the user to a page where the topic and all its characteristics are shown. The page uses the TopicExtractorQuis class that has just been described.

```
// Importing the topic map application Java module
<%@ page import="tm4jweb.ferdig.*" %>

// A cookie is used for common data shared by the JSP pages
<jsp:useBean id="cookie1" class="tm4jweb.ferdig.tmcookie" scope="session"/>
<jsp:setProperty name="cookie1" property="a"/>
<jsp:setProperty name="cookie1" property="b"/>
<jsp:setProperty name="cookie1" property="c"/>

// The HTML code begins
<html>

    // Page layout with color and logo
    <body bgcolor="#aaaaee">
    
    <br/><br/><br/>
    <font size=3><b>QQuality, Interoperability and Standards e-learning</b></font>
    <br/><br/>

    // A link back to the home page of the application
    <a href="quis.jsp">home</a>

    // A table listing all the topics
```

Implementation and documentation

```
<center>
<br/><br/><br/><font size=3><b>All Topics</b></font></br>
<table frame="box" border="2">

  <%
    // The class described in the previous sub-chapter
    TopicExtractorQuis teq = new TopicExtractorQuis();

    // Importing the topic map
    teq.importer("c:/diplom/tm4j-
0.9.7/examples/topicmaps/kings_and_queens.xtm");

    // Collecting the topics with the getTopic() method, also described
    // in the last sub-chapter.
    String[] topicsArray = teq.getTopics();

    // A dynamical loop that prints one array row for each topic.
    int length = java.lang.reflect.Array.getLength(topicsArray);
    out.print("<br/>");
    for(int i=0;i<length;i++){
  <%>

  <tr>
  <td valign="center" align="center" bgcolor="#dddddd">

    // A link for each of the topic names. The link refers to a
    // page called topic.jsp, and takes the topic name as a parameter.
    // The topic.jsp page shows the topic with all its characteristics.

    <a href="topic.jsp?a=
  <%
      out.print(topicsArray[i]);
  <%>
  &c=topic">
  <%
      out.print(topicsArray[i]);
      out.print("</a>");
      out.print("<br/>");
  <%>
  </td></tr>
  <%
    }
  <%>

</table>

</center>
</br>

</body>
</html>
```

6.2 Installation Guide

This is the system documentation. The chapter describes how to install the systems.

6.2.1 Topic map constructor system

The topic map constructor system consists of six Java classes. In order to run the system, the following has to be done:

- Java SDK must be installed on the computer. The system has been tested with version 1.4.X and 1.5.X.
- Copy the Java files to a local directory and add the directory to the Java classpath.
- Download and install TM4J. Installation files and guide can be found at www.tm4j.org. Make sure the tm4j installation is added to the classpath.
- Compile the Java files using any Java compiler.
- The topic map is written to a directory called c:\temp\, and the name of the file is “test.xtm”. If a different path or file name is wanted, these can be changed in the Java source files prior to compilation.
- In order to run the system, a file called CatalogManager.properties is needed. Create the file and add the text “catalogs=”. This file must also be placed in a directory present in the classpath.
- Run the system in a console window by typing “java MainClass”.

6.2.2 Topic map application system

The topic map application system consists of five Java files and seven JSP files. The system might be run on several different web application servers. This guide will explain how to install and run it under Tomcat version 5.5. The following has to be done:

Installing Tomcat

- Download and install Tomcat version 5.X from <http://tomcat.apache.org/download-55.cgi>.
- Chose which port to use (default: 80).

Implementation and documentation

- The Tomcat installation contains a directory called “webapps”. Create a new directory under “webapps” and call it “tm4j”.

Installing TM4J in Tomcat

- Copy all the JAR files in the TM4J library to the sub-folder “shared/lib” under the Tomcat installation. Add this directory to the classpath.
- Make a new JAR file containing the CatalogManager.properties file and add this to the “shared/lib” folder. The CatalogManager.properties file will not be recognized unless it is in a JAR file.

Installing the JSP and Java files

- Copy all the JSP files to the “tm4j” directory.
- Create a new directory under “tm4j” called “WEB-INF/tm4jweb/ferdig”.
- Add the Java files to this directory.
- The files can be added to any other directory under WEB-INF. If this is desirable, the package declaration within the Java files must be changed prior to compilation.
- Compile the Java files.
- Start the Tomcat server.
- The application is available via a web browser at `http://{server_address}:{port}/tm4j/quis.jsp`. If you for instance run it from the pc it is installed and use port 8080, the address would be:
<http://localhost:8080/tm4j/quis.jsp>.

6.3 User's Guide

This is the user documentation for the QUIS topic map constructor system. It describes how to start the system and add topic elements.

6.3.1 Starting the system

The system must be started in a console window from the directory where it is installed. It is started by running the class called "MainClass".

6.3.2 Importing or creating topic maps

The system does currently not have a menu for importing topic maps.

6.3.3 Select element to be added

The first menu that appears when the system is run lets the user select which type of topic map element he wants to add: Topic, Association or Occurrence. The menu is shown in Figure 16.

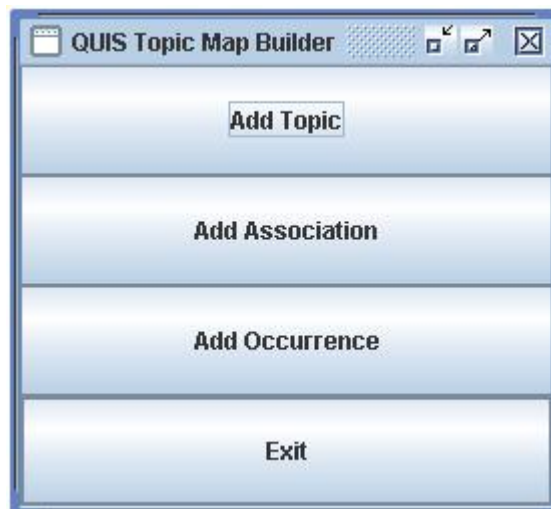


Figure 16 Select topic map element

6.3.4 Add topic

When "Add Topic" is chosen, a new menu appears, shown in Figure 17.

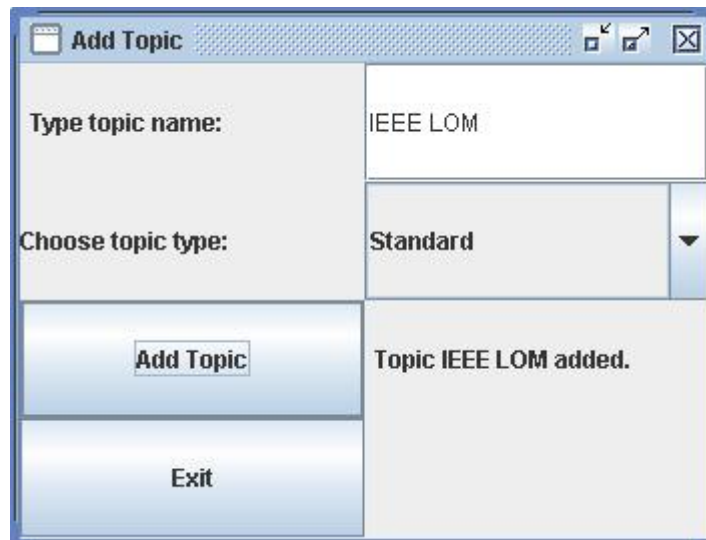


Figure 17 Add Topic

Type the new topic's name in the text field behind "Type topic name". Choose the topic's type from the combo box below. If no type is wanted, this field can be left blank. Press the "Add Topic" button to add the topic to the topic map. A message will confirm the operation, or present a potential error.

6.3.5 Add association

When "Add association" is chosen the menu shown in Figure 18 appears.

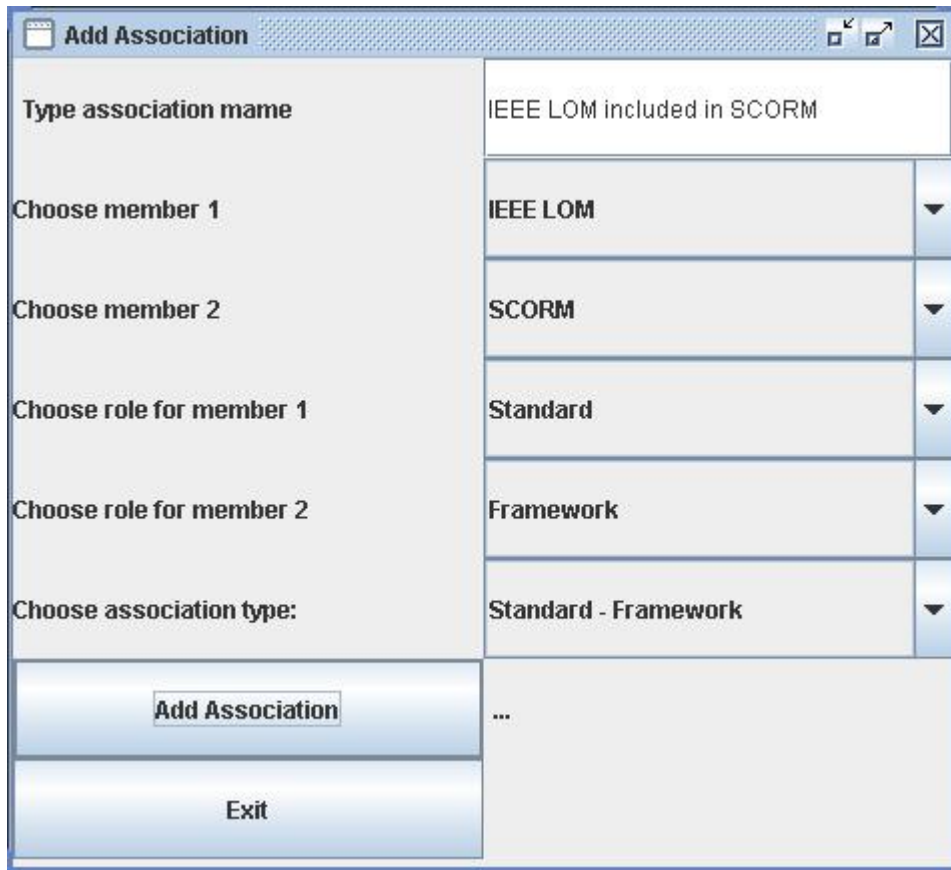


Figure 18 Add association

In the upper text field the association's name is written. The program only supports association between two topics. These topics must be chosen in the two topmost combo boxes. The topics in an association are called the associations' members. The members play different roles in the association. These roles are chosen in the subsequent combo boxes. The last combo box lets the user choose an association type. Finally the user adds the association to the topic map with a button.

6.3.6 Add occurrence

When "Add occurrence" is chosen the menu shown in Figure 19 appears.

Type occurrence:	http://www.adlnet.org/
Select topic for occurrence:	SCORM
Select occurrence type:	Web page
Select internal og external occurrence:	External
Add Occurrence	
Exit	
Occurrence added...	

Figure 19 Add Occurrence

In the upper text field the occurrence is typed. For internal occurrences, this means information about the subject. For external occurrences it means a reference to external information about the subject. The occurrence in Figure 19 is external. This is chosen in the last combo box. The occurrence' topic is "SCORM". The type of the occurrence is chosen in a combo box, and in this case it is the type "web page". The occurrence typed in is the home page of the organization that has developed SCORM.

7 Test document

The purpose of this document is to describe how the system should be tested, to achieve the most optimal result. During the development of a system, errors of different degrees will occur no matter how well you work. Thus testing is needed to discover and correct these errors.

Testing of a system like this has two purposes; to validate and to verify the system. Validation and verification are described in the bulleted list below.

- *Validation* of a system means to ensure that the system offers and carries out the intended functionality. In other words “to ensure that the correct system is being made”. The validation is based on the requirements in the requirements specification document.
- *Verification* of a system means to ensure that the programming code is free of errors and to ensure that the system is stable. In other words “to ensure that the system is being made correctly.”

Chapter 7.1 describes the different test principles and methods that are used. Thereafter follows the different tests chapter 7.2. Chapter 7.3 contains the test results. A final conclusion is given in chapter 7.4.

7.1 Test principles and methods

This chapter presents the different test principles and methods used during testing. These are first discussed in general, and then how and when they will be used during this system development.

7.1.1 “White box” testing

“White box” testing means to analyze code to detect and eliminate errors. These tests are separated into static and dynamic analysis, which are explained in the next sub-chapters.

Static analysis

Static analysis does not require the program to run while testing. Figure 20 gives a survey over methods of static analysis.

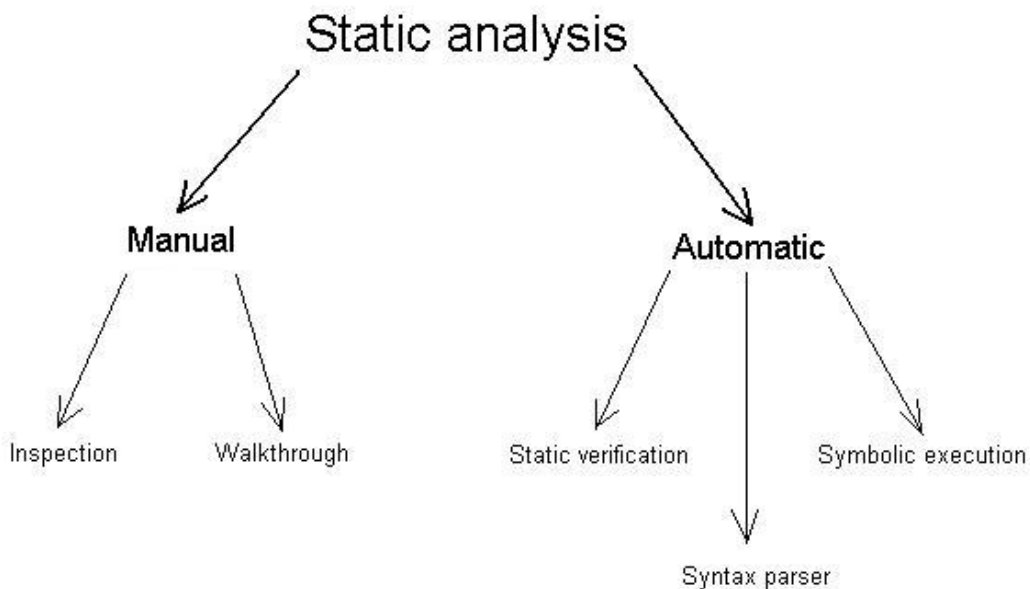


Figure 20 Static analysis

Automatic static analysis is to compare the program specification with the written code. It is used to ensure that the program actually does everything it is meant to do. This type of testing requires a formalized specification in order for using formal methods to prove system

functionality. This type of testing does not however test whether the program does what it is meant to do, that is; whether the specification is correct. Considering the size of the system, the time used to develop these types of tests is not reckoned to be justifiable. Automatic static analysis will therefore not be used.

Manual inspection, or code inspection, is a common way of testing programming code. Normally small test teams are put together and walk thoroughly through the code. The code should preferably be distributed a day in advance, so that the team members who have not written the code have time to prepare for the inspection. During the inspection the code is systematically examined line by line, to detect errors and contradictions. Manual inspection will be used for unit testing, by using pre-defined checklists for code inspection. As the system is made by one person only, no team will be put together. The system developer will do all the testing.

A manual walkthrough is an even more formal examination of the code. A walkthrough is done by a test team including the code writer, a secretary and a moderator. This type of testing is also reckoned to be too excessive with a system of this size, and will therefore not be used.

Dynamic analysis

In a dynamic analysis the program is run while testing. The purpose of this is to understand the loops and branches done during program execution. This type of testing requires a pre-defined set of input data to the system. Dynamic analysis is therefore much like “black box” testing, which will be described in the next chapter. The advantage of using dynamic analysis is that it shows how the program produces output data from the input data. It will not however be used during this system development as it is too time consuming.

7.1.2 “Black box” testing

“Black box” testing does not concern how the program executes. It merely looks at input and output data. The program code becomes invisible to the person running the test, therefore the name “black box” testing. Figure 21 shows the principles of “black box” testing. The functional requirement to the system states what the program should do, and not how. “Black box” testing is therefore a useful approach, and will be used.

"Black box" testing



Figure 21 "Black box" testing

A prerequisite for this test to be useful is that all code errors are removed. Otherwise the system might stop or loop during the test, and no output will be produced at all.

7.1.3 Checklists

A checklist is a bulleted list where each bullet describes an action for detecting and removing typical programming errors. Checklists will be used to test the Java and JSP code during unit testing.

7.2 Tests

This chapter presents the actual tests. The tests will be separated into different sub-chapters depending on the nature of the test and the requirements being tested.

7.2.1 Testing requirements concerning documentation

In this chapter one test is introduced. It tests the requirements concerning the documentation of the system.

Test 1 Test of documentation

Name:	T1 Test of documentation
Consumption of time:	30 min
Requirements tested:	NFR-UD01 The user documentation must be written in English.

Test document

	<p>NFR-SD02 The system documentation must be written in English.</p> <p>NFR-CD01 The code documentation must be written in English.</p> <p>NFR-TD01 The test documentation must for each test explain which requirements the test is verifying.</p> <p>NFR-UD02 The user documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.</p> <p>NFR-SD01 The installation documentation should explain how to install the system.</p> <p>NFR-SD03 The system documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.</p> <p>NFR-CD02 All code should be documented with Javadoc.</p> <p>NFR-TD02 Each variant of testing should be explained, and the use of the variant should be reasoned for.</p>
--	--

7.2.2 Testing the graphical user interface

Testing all the requirements concerning the graphical user interface is also done in one test.

Test 2 Test of graphical user interface

Name:	T2 Test of graphical user interface
Consumption of time:	30 min
Requirements tested:	<p>GR-02 Input to the system must be done with the mouse, the keyboard and via file input.</p> <p>GR-03 Response from the system must be visual on screen.</p> <p>GR-01 The user should interact with the system via links and search bars.</p> <p>GR-04 The system should not present any “surprising” elements.</p> <p>GR-05 The system should have a maximum of seven new elements on each screen level.</p>

7.2.3 Testing requirements concerning implementation

Test 3 Test of implementation

Name:	T3 Test of implementation
Consumption of time:	30 min
Requirements tested:	<p>NFR-I02 All programming must be done in English.</p> <p>NFR-I03 The implementation must be as modular as possible.</p> <p>NFR-I01 The system should be made open source.</p> <p>NFR-I04 All written code should be commented and explained.</p>

7.2.4 Testing requirements concerning software, accessibility and security

The testing of the requirements concerning software, accessibility and security is summarized in one single test.

Test 4 Test of software, accessibility and security

Name:	T4 Test of software, accessibility and security
Consumption of time:	30 min
Requirements tested:	<p>NFR-SW01 The system must be accessible by all web browsers supporting JSP.</p> <p>NFR-A01 The system must be accessible via the www.tisip.no/quis web page.</p> <p>NFR-A02 The maximum response time of the system must under normal circumstances be less than three seconds.</p> <p>NFR-S01 Access to the “Partners’ arena” must require a username and a password.</p> <p>NFR-A03 When the response time exceeds three seconds the user should receive a message.</p>

7.2.5 Testing functional requirements for the topic map constructor system

The functional requirements for the topic map constructor system are tested in one single test.

Test 5 Test of the functional requirements for the topic map constructor system

Name:	T5 Test of functional requirements for the topic map constructor system.
Consumption of time:	30 min
Requirements tested:	<p>FRS1-01 It must be possible to add topics to the topic map.</p> <p>FRS1-02 It must be possible to type the topics, associations and occurrences in the topic map.</p> <p>FRS1-03 It must be possible to add associations with members and member roles to the topic map.</p> <p>FRS1-04 It must be possible to add occurrences to the topic map.</p> <p>FRS1-05 It should be possible to add scopes to the topic map elements.</p>

7.2.6 Testing functional requirements for the topic map application system

The last test concerns the functional requirements of the topic map application system.

Test 6 Test of the functional requirements for the topic map application system

Name:	T6 Test of the functional requirements for the topic map application system.
Consumption of time:	30 min
Requirements tested:	<p>FRS2-01 The system must give the user access to all topics.</p> <p>FRS2-02 The system must give the user access to all topic types.</p> <p>FRS2-03 The system must give the user access to all associations.</p> <p>FRS2-04 The system must give the user access to all association types.</p> <p>FRS2-05 The system must give the user access to single topics.</p> <p>FRS2-06 The system must give the user access to single</p>

	<p>associations.</p> <p>FRS2-07 The system must give the user access to relevant topic information.</p> <p>FRS2-08 It should be possible to search the topic map.</p>
--	---

7.3 Test results

In this sub-chapter are the test results presented. On table is filled in for each test.

Test report for: T1 Test of documentation

Name:	T1 Test of documentation
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes and Geir Ivar Holberg
Consumption of time:	60 min
Requirements tested:	<p>NFR-UD01 The user documentation must be written in English.</p> <p>NFR-SD02 The system documentation must be written in English.</p> <p>NFR-CD01 The code documentation must be written in English.</p> <p>NFR-TD01 The test documentation must for each test explain which requirements the test is verifying.</p> <p>NFR-UD02 The user documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.</p> <p>NFR-SD01 The installation documentation should explain how to install the system.</p> <p>NFR-SD03 The system documentation should be written in a way that it can be read and understood by a person having English as a second language within 30 minutes.</p> <p>NFR-CD02 All code should be documented with Javadoc.</p> <p>NFR-TD02 Each variant of testing should be explained, and the use of the variant should be reasoned for.</p>
Results:	<p>All documentation is written in English.</p> <p>The test documentation explains which requirements the test is verifying.</p>

Test document

	<p>The test person Geir Ivar Holberg who has no knowledge of topic maps used less than 30 minutes to read and understand the system and user documentation.</p> <p>He also verifies that the system documentation explains how to install the system.</p> <p>All code is not yet documented with Javadoc.</p> <p>Each variant of testing is explained and reasoned for.</p>
Required actions:	The test shows that Javadoc must be written during the last week of the project.

Test report for: T2 Test of graphical user interface

Name:	T2 Test of graphical user interface
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes
Consumption of time:	30 min
Requirements tested:	<p>GR-02 Input to the system must be done with the mouse, the keyboard and via file input.</p> <p>GR-03 Response from the system must be visual on screen.</p> <p>GR-01 The user should interact with the system via links and search bars.</p> <p>GR-04 The system should not present any “surprising” elements.</p> <p>GR-05 The system should have a maximum of seven new elements on each screen level.</p>
Results:	All requirements were satisfied except GR-01 . The system does not contain any search functionality.
Required actions:	Due to time shortage there will not be implemented any search functionality.

Test document

Test report for: T3 Test of implementation

Name:	T3 Test of implementation
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes
Consumption of time:	30 min
Requirements tested:	<p>NFR-I02 All programming must be done in English.</p> <p>NFR-I03 The implementation must be as modular as possible.</p> <p>NFR-I01 The system should be made open source.</p> <p>NFR-I04 All written code should be commented and explained.</p>
Results:	<p>Requirements NFR-I02 and NFR-I01 are satisfied.</p> <p>The system developer feels that requirement NFR-I03 is satisfied, although this is a matter of opinion</p> <p>The requirement NFR-I04 is not completely satisfied due to missing Javadoc.</p>
Required actions:	The test also shows that Javadoc must be written during the last week of the project.

Test report for: T4 Test of software, accessibility and security

Name:	T4 Test of software, accessibility and security
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes
Consumption of time:	30 min
Requirements tested:	<p>NFR-SW01 The system must be accessible by all web browsers supporting JSP.</p> <p>NFR-A01 The system must be accessible via the www.tisip.no/quis web page.</p> <p>NFR-A02 The maximum response time of the system must under normal circumstances be less than three seconds.</p> <p>NFR-S01 Access to the "Partners' arena" must require a username and a password.</p> <p>NFR-A03 When the response time exceeds three seconds the user</p>

Test document

	should receive a message.
Results:	<p>The topic map application system has been tested with Internet Explorer and Netscape.</p> <p>The system is not yet available on the www.tisip.no/quis web page.</p> <p>The system response time is within the limits.</p> <p>There is no password protection implemented.</p>
Required actions:	<p>The system must be implemented on the correct location.</p> <p>Password protection must be enabled before any sensitive data can be added to the topic map.</p>

Test report for: T5 Test of the functional requirements for the topic map constructor system

Name:	T5 Test of the functional requirements for the topic map constructor system.
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes
Consumption of time:	30 min
Requirements tested:	<p>FRS1-01 It must be possible to add topics to the topic map.</p> <p>FRS1-02 It must be possible to type the topics, associations and occurrences in the topic map.</p> <p>FRS1-03 It must be possible to add associations with members and member roles to the topic map.</p> <p>FRS1-04 It must be possible to add occurrences to the topic map.</p> <p>FRS1-05 It should be possible to add scopes to the topic map elements.</p>
Results:	The only requirement not satisfied is FRS1-05 .
Required actions:	Functionality for adding scopes to the topic map elements should be implemented.

Test document

Test report for: T6 Test of the functional requirements for the topic map application system

Name:	T6 Test of the functional requirements for the topic map application system.
Date:	16 Jan 2006
Carried out by:	Frode Hjeltnes
Consumption of time:	30 min
Requirements tested:	FRS2-01 The system must give the user access to all topics. FRS2-02 The system must give the user access to all topic types. FRS2-03 The system must give the user access to all associations. FRS2-04 The system must give the user access to all association types. FRS2-05 The system must give the user access to single topics. FRS2-06 The system must give the user access to single associations. FRS2-07 The system must give the user access to relevant topic information. FRS2-08 It should be possible to search the topic map.
Results:	The only requirement not satisfied is FRS2-08 .
Required actions:	Search functionality should be implemented in the topic map application system.

7.4 Test conclusion

The test results show that six requirements are not satisfied. Two of these are of high importance:

- **NFR-A01** The system must be available via the www.tisip.no/quis web page.
- **NFR-S01** Access to the “Partners’ arena” must require a username and a password.

The system can not be published for common use until there requirements are fulfilled.

The remaining requirements which are not fulfilled are:

Test document

- **NFR-I04** All written code should be commented and explained.
- **NFR-CD02** All code should be documented with Javadoc.
- **FRS1-05** It should be possible to add scopes to the topic map elements.
- **FRS2-08** It should be possible to search the topic map.

The two first ones both fail due to missing Javadoc. Since these four requirements are of medium importance, their fulfillment will be prioritized after the requirements of high importance.

8 Results and discussion

This thesis does not contain any scientific formulas or theorems to be tested. Nor does it contain any hypothesis to be accepted or rejected. The results consist of the actual systems that are implemented. And what's more important, an evaluation of weather topic maps actually provides any advantages compared to existing web portals. This evaluation or discussion will be given along with the presentation of the implemented systems.

All of the requested system functionality is not implemented. This chapter will base its discussion on the functionalities that are implemented. These are after all the results. How the developed systems can or should be extended, and which potential results this might give are presented in chapter 9 "Further work".

This chapter first presents results concerning the topic map constructor system. Then the topic map application system is discussed. Finally a discussion of the TM4J library is presented. The student spent so much time with this library that some comments should be made.

8.1 Results from the topic map constructor system

The topic map constructor system gives the user the possibility to build a topic map with topics, associations and internal and external occurrences. The interface is plain and simple. To be able to use the system, the user must possess some basic knowledge of topic maps. The topic map made in the system follows the XTM format, specified in the topic map standard.

The topic map constructor system will not be used by persons looking for information on the Internet. It still provides answers to weather topic map applications are favorable compared to regular web portals. Regular web portals also need administrators who update the web pages and add and remove links and pages. As the portal grows large, the web of links and pages becomes increasingly difficult to manage. With the topic map constructor system it is practically just as easy to add topic number 10 as topic number 10000. This is because the entire topic map is one XML file, and the web pages are generated dynamically upon request.

8.2 Results from the topic map application system

The topic map application system consists of Java and JSP files that are used by a Tomcat web application server to present topic map information via web browsers. The system gives the user access to the main topic map elements such as topics, associations and occurrences, as well as the links between them.

8.2.1 Topic maps can improve web portals

The main objective for the thesis was to examine the topic maps' strength as an information provider, compared with regular web portals.

Even though both the topic map and the topic map application made in this project are slightly less functional than optimal, the conclusion is clear. By comparison with other well known e-Learning portals it is easy to see the advantageous of the topic map application. The most evident observation is the simplicity in finding information about e-Learning standardization and nothing more than that. Information about everything else is cut out. The application works as a reference book for finding requested information, in addition to containing valuable data of its own.

8.2.2 Easy to maintain and extend

Extension and further development of regular web portals is bothersome work. All new pages needs to be manually crafted with information and links. All other pages that shall contain a link to the new page must also be modified. Removing a page from the portal has the same effects. Every page that contains a link to the removed page must be updated and the link must be removed. Otherwise you end up with a system with dead end dangling links.

With a topic map all that is needed is to add or remove instances within the topic map. The topic map application then dynamically generates the web pages. Updating the topic map is done with a system like the topic map constructor system developed in this project. A person with no knowledge of topic maps would not know how to do this. Some kind of topic map administrator is therefore needed. With such a person present however, the maintenance is easily done. The problem of dead end links still remains if the dead link is to an information source outside the topic map, that is; an external occurrence.

8.2.3 May use common interfaces

One of the problems of introducing new technology is that people in general are afraid of it. With topic maps this is not a problem. Topic map applications can be run with a common web browser, as demonstrated in this project. They may also be independent Java applications or applications written in any other programming language.

8.3 Discussion concerning TM4J

The main reason why there was not enough time to complete the topic map systems, is the size and complexity of TM4J. The library contains a very large amount of classes and interfaces, which are not always strongly documented. It took several weeks to become familiar with how the library interprets the topic map standard and translates it into Java classes and method, even though the student is experienced both within the fields of Java development and topic maps.

When the student finally became comfortable with TM4J, it became obvious that it is a strong implementation. It is very general and modular and includes a large amount of interfaces rather than finished classes. This makes the system harder to learn, but easier to use when you first get to know it.

Additional comments and documentation would lower the threshold for learning the system. A complete, implemented and documented example of how the system can be used would also be preferable. When the QUIS topic map systems are finalized they might provide such an example.

9 Further work

The desired amount of results is not obtained due to the lack of implemented system functionality. This is a result of an insufficient amount of programming time. This chapter discuss' what can and should be done to finalize and possibly extend the systems. Thereafter some comments are made on additional work that can be done.

9.1 System enhancement

This sub-chapter presents the changes and further work that has to be done on the QUIS topic map systems. Common for both systems is lacking Javadoc. Due to the lack of some important functionality, the Javadoc is not written. This has to be done when the systems are finalized.

9.1.1 Topic map constructor system

The topic map constructor system does not yet provide the functionality of adding scopes to the topic map elements. The use of scopes is discussed in detail in chapter 2.5, and is considered to be one of the major strengths of topic maps. The possibility to add scopes could with advantage be build into the system. One of the settings where scopes could be used is when a topic has multiple names. Adding multiple names to a topic is also something the system should provide functionality for.

When the topic map administrator is done updating or creating a topic map, the system writes the topic map as an XML file, following the XTM standard. According to the topic map standard, the HyTime (HyTM) format can also be used. The topic map constructor system might be extended to provide functionality of writing the topic map to both formats.

Other features like for instance merging of topic maps could be included in the system. An already implemented application for merging topic maps is included in the TM4J library. This functionality could therefore easily be included in the topic map constructor system.

Another thing that is lacking is the possibility to remove topic elements from a topic map. When information no longer exist, is outdated or is wrong, the topic map administrator must have the possibility to remove topics, associations, occurrences and scopes.

Further work

Finally the functionality for importing a topic map to the system must be added.

9.1.2 Topic map application system

It remains a lot of work to be done on the topic map application system. Lacking functionality is due to shortage of development time.

The system can be extended to include functionality in the following areas:

- **Security**

Requirement **NFR-S01** states that topic map elements under the scope “Partners’ arena” should be protected with a username and a password. Scoping topic map elements is not possible with the topic map constructor system as it is today. This might change in the future, so username and password protection should be implemented in the topic map application system.

- **Scope**

The system can be extended to handle scoped elements.

- **Expansion testing**

The system has no functionality for load sharing. Today the system response time is within the requested area when used by up to five users simultaneously. The system has not been tested with more users. A potential slow down must be taken into account and tested for. As the system and the topic map potentially grows this becomes more relevant.

- **Searching**

Search functionality is something that absolutely should be included. One of the strengths with topic maps is that the user easily can find relevant information without searching. Searching is still something that users are familiar with, and would represent a considerable value to the overall system.

Further work

Figure 13 in the systems design document shows a class diagram of the topic map application system Java module. In this figure a file called “QuisTopicMap.java” is shown. This file is intentionally not made, as it was clear during implementation that there was no need for it.

9.2 Additional work

This sub-chapter discuss' some final work that can be done with the system.

9.2.1 Documentation

A strong documentation is a necessity for other user to continue working with the code. System documentation must be made regardless of weather the system developer continues to work with the code himself. Both systems' documentation can be improved. The User's Guide does not yet contain documentation of the topic map application system. This must be added when the system is finalized.

9.2.2 PSI

The topics in the topic map created in this project do not use PSIs. The use of PSIs can enable and encourage merging of topic map within the area of e-Learning standardization. There are many different, and to a certain degree competitive, parties within this area. PSIs should be published on a stable web site and used in the topic map made in this thesis. At the moment of writing there exist no such PSIs.

9.2.3 Open source publication

The system can be published to an open source topic map community. This will give other topic map developers the opportunity to extend and enhance the system on their own. The more persons who use the system, the more persons also use the PSIs. This is an additional encouragement for publishing PSIs.

10 References

These references lead to supplementary information within the following areas:

- Topic maps
- QUIS
- TM4J
- Technologies

10.1 *Topic maps*

The following links are references to information about topic maps:

- <http://www.topicmap.com/> - The official topic map web page
- <http://www.ontopia.net/topicmaps/materials/tao.html> - The TAO of topic maps
- http://www.empolis.com/downloads/empolis_TopicMaps_Whitepaper20030206.pdf - The topic maps hand book
- <http://www.infoloom.com/xtmguide.htm> - The XTM guide
- <http://www.infoloom.com/tnm/draft27.pdf> - The ISO/IEC 13250 Topic Maps standard

10.2 *QUIS*

The following links are references to information about QUIS:

- <http://www.tisip.no/quis> - The official QUIS web page
- http://www.elearningeuropa.info/index.php?page=doc&doc_id=6043&doclng=6&menuzone=1 – The EU e-Learning program’s description of the project

10.3 *TM4J*

The following link lead to information about TM4J:

- <http://tm4j.org/> - The TM4J home page

10.4 Technologies

The following links are references to different tools and technologies discussed in this thesis:

- <http://www.tmap.org/> - The Topic Map API
- <http://www.php.net/> - The Hypertext Preprocessor
- <http://java.sun.com/products/jsp/> - The Java Server Pages
- <http://jakarta.apache.org/tomcat/> - Tomcat
- <http://cocoon.apache.org/2.1/> - The Cocoon Project

Appendix A: ISO/IEC 13250 Topic Maps

13250

First

December 3, 1999

E

ISO/IEC 13250

Topic Maps

Information Technology

Document Description and Processing Languages

Technologies de l'information -- Cartes topiques

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 13250 was prepared by Joint Technical Committee 1 JTC1, Information technology, Subcommittee SC34, Document description and processing languages.

Introduction

This International Standard provides a standardized notation for interchangeably representing information about the structure of information resources used to define topics, and the relationships between topics. A set of one or more interrelated documents that employs the notation defined by this International Standard is called a *topic map*. In general, the structural information conveyed by topic maps includes:

- groupings of addressable information objects around topics (‘occurrences’), and
- relationships between topics (‘associations’).

A topic map defines a multidimensional topic space — a space in which the locations are topics, and in which the distances between topics are measurable in terms of the number of intervening topics which must be visited in order to get from one topic to another, and the kinds of relationships that define the path from one topic to another, if any, through the intervening topics, if any.

NOTE 1 Two topics may be connected through an association, and they can also be connected by virtue of sharing an occurrence.

In addition, information objects can have properties, as well as values for those properties, assigned to them externally. These properties are called *facet types*.

NOTE 2 The word *facet* can mean one side of a many-sided, polished object, or one segment of a compound eye (e.g. an insect's). Its metaphorical use here captures the idea that a facet is a property of a set of information objects that can be used to create a view of them.

Several topic maps can provide topical structure information about the same information resources. The Topic Maps architecture is designed to facilitate merging topic maps without requiring the merged topic maps to be copied or modified. Because of their extrinsic character, topic maps can be thought of as *overlays* on, or extensions to, sets of information objects.

The base notation of Topic Maps is SGML; an interchangeable topic map always consists of at least one SGML document, and it may include and/or refer to other kinds information resources. A set of information resources that comprise a complete interchangeable topic map can be specified using the ‘bounded object set (BOS)’ facility defined by the HyTime architecture in ISO/IEC 10744:1997.

As the Extensible Markup Language (XML), a World Wide Web Consortium recommendation, is a subset of SGML, as explained in Annex K of SGML (1997), also known as WebSGML, XML can be also used as a base notation for Topic Maps.

The topic map notation is defined as an *SGML Architecture*, and this International Standard takes the form of an *architecture definition document* expressed in conformance with Normative Annex A.3 of ISO/IEC 10744:1997, the *SGML Architectural Form Definition Requirements (AFDR)*. The formal definition of the topic map notation is expressed as a meta-DTD.

ISO/IEC 13250, Topic Maps

1. Scope	1
2. Normative references	1
3. Definitions	2
3.1 added themes	2
3.2 association	2
3.3 association link	2
3.4 association role	2
3.5 association type	2
3.6 base name	2
3.7 bounded object set (BOS)	3
3.8 display name	3
3.9 facet	3
3.10 facet link	3
3.11 facet type	3
3.12 facet value	3
3.13 hub document	3
3.14 occurrence role	3
3.15 public subject descriptor	3
3.16 scope	3
3.17 sort name	4
3.18 subject	4

3.19 subject descriptor	4
3.20 theme	4
3.21 topic	5
3.22 topic association	5
3.23 topic characteristic	5
3.24 topic characteristic assignment	5
3.25 topic link	5
3.26 topic map	5
3.27 topic name	6
3.28 topic occurrence	6
3.29 topic type	6
3.30 unconstrained scope	6
4. Notation	6
4.1 RCS name, full name, description, and clause	7
4.2 Constraints	7
4.3 Note	8
5. Topic Maps architecture	8
5.1 Topic Map Architectural Form	8
5.2 Topic link	10
5.2.1 Topic Link Architectural Form	10
5.2.2 Topic Name Architectural Form	12
5.2.3 Topic Occurrence Architectural Form	15
5.3 Association Link	18
5.3.1 Association Link Architectural Form	18
5.3.2 Association Role Architectural Form	19

5.4 Themes To Be Added Architectural Form	21
5.5 Facet Linking	22
5.5.1 Facet Link Architectural Form	23
5.5.2 Facet Value Architectural Form	24
6. Conformance	25
Annexes	27
A. Annex A: Topic Maps Meta-DTD (Normative)	27
B. Annex B: Example Architectural Support Declaration for the Topic Maps Architecture (Informative)	39

1 Scope

NOTE 1 This clause defines the scope of this International standard. It should not be confused with the concept of "scope" defined in 3.16, which only applies in the context of Topic Maps.

Topic maps enable multiple, concurrent views of sets of information objects. The structural nature of these views is unconstrained; they may reflect an object oriented approach, or they may be relational, hierarchical, ordered, unordered, or any combination of the foregoing. Moreover, an unlimited number of topic maps may be overlaid on a given set of information resources.

Topic maps can be used:

- To qualify the content and/or data contained in information objects as topics to enable navigational tools such as indexes, cross-references, citation systems, or glossaries.
- To link topics together in such a way as to enable navigation between them. This capability can be used for virtual document assembly, and for creating thesaurus-like interfaces to corpora, knowledge bases, etc.
- To filter an information set to create views adapted to specific users or purposes. For example, such filtering can aid in the management of multilingual documents, management of access modes depending on security criteria, delivery of partial views depending on user profiles and/or knowledge domains, etc.
- To structure unstructured information objects, or to facilitate the creation of topic-oriented user interfaces that provide the effect of merging unstructured information bases with structured ones. The overlay mechanism of topic maps can be considered as a kind of *external markup mechanism*, in the sense that an arbitrary structure is imposed on the information without altering its original form.

This International Standard does not require or disallow the use of any scheme for addressing information objects. Except for the requirement that topic map documents themselves be expressed using SGML (or WebSGML) and HyTime, using the syntax described herein, neither does it require or disallow the use of any notation used to express information.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8879:1986, *Information processing — Text and office systems — Standard Generalized Markup Language (SGML)*, including Technical Corrigenda 1, 2, and 3.

ISO 10744:1997, *Information processing — Text and office systems — Hypermedia/Time-based Structuring Language (HyTime)*, including Amendment 1.

3 Definitions

For the purposes of this International Standard the following definitions apply.

The definitions provided in ISO 8879:1986 (including TC1, TC2 and TC3) and ISO/IEC 10744:1997 (including Amendment 1) shall apply to this International Standard.

3.1 added themes

Topics added to the sets of themes comprising the scopes within which topics have their topic characteristics. Added themes can be specified in two ways:

- a) Within the topic map document whose scopes are affected, by means of the **added themes** (*addthemes*) attribute of the document element. The specified themes are added to the scopes of all of the topic characteristics which are assigned to topics via the topic links and association links contained in the document.
- b) Inside or outside the topic map document whose scopes are affected, by means of elements conforming to the **themes to be added** (*addthms*) architectural form. The specified themes are added to the topic characteristics assigned to topics via:
 - entire topic map documents (specified via the *tmdocs* attribute),
 - topic links (that is, the name characteristics and occurrence characteristics assigned to topics via topic links) (specified via the *cassign* attribute),
 - association links (that is, the roles played in associations by topics, as assigned to topics via association links) (specified via the *cassign* attribute), or
 - any combination of the foregoing.

3.2 association

See 'topic association'.

3.3 association link

A hyperlink element conforming to the association link architectural form defined by this International Standard. See 5.3.

3.4 association role

One of the roles that topics play in a given topic association.

3.5 association type

- a) A subject which is a class of topic associations.
- b) One of the classes of topic associations of which a particular association link is an instance. The association types of which a given association link is an instance can be specified by its optional *types* attribute.

3.6 base name

- a) A subelement (*basename*) of a *topname* subelement of a topic link.

b) A name characteristic of a topic that is specified in the content of a *basename* element.

3.7 bounded object set (BOS)

A set of one or more documents and other information objects, all of which are known to the processing application and which are processed collectively. See ISO/IEC 10744:1997 for details; see also the definition of ‘hub document’.

3.8 display name

a) A subelement (*dispname*) of a *topname* subelement of a topic link, containing the identifying information intended to be displayed by the application to represent the subject of the topic link.

b) A name characteristic of a topic that is specified in the content of a *dispname* element.

3.9 facet

a) The subset of information objects that share an externally-applied property.

b) The values given to a particular property externally applied to a set of information objects.

3.10 facet link

A hyperlink that applies values for a given property (as well as the property itself) to one or more information objects.

3.11 facet type

A property applied by one or more facet links to one or more objects.

3.12 facet value

A member of the set of all values of a particular facet type.

3.13 hub document

The HyTime document used to define the set of information resources (the *bounded object set* (BOS)) that comprise a HyTime hyperdocument. Applications may regard the HyTime document used as the entry point for a browsing session within a hyperdocument as the hub document. See ISO/IEC 10744:1997 for details. By definition, a topic map is a HyTime hyperdocument, and any topic map document can be regarded as a hub document.

3.14 occurrence role

The sense in which some set of occurrences is relevant to a topic. In the Topic Maps architecture, occurrence roles are specified as anchor roles (as defined in the HyTime architecture) of topic links.

3.15 public subject descriptor

A subject descriptor (see the definition of ‘subject descriptor’) which is used (or, especially, which is designed to be used) as a common referent of the *identity* attributes of many topic links in many topic maps. The subject described by the subject descriptor is thus easily recognized as the common binding point of all the topic links that reference it, so that they will be merged.

3.16 scope

NOTE 2 The definition of scope provided here should not be confused with 1, entitled ‘Scope’, which defines the scope of this International Standard.

The extent of the validity of a topic characteristic assignment (see the definition of ‘topic characteristic assignment’): the context in which a name or an occurrence is assigned to a given topic, and the context in which topics are related through associations. This International Standard does not require that scopes be specified explicitly. If the scope of a topic characteristic assignment is not explicitly specified via one or more *scope* attributes, the scope within which the topic characteristic applies to the topic includes all the topics in the entire topic map; this special scope is called ‘the unconstrained scope’. If a scope is specified, the specification consists of a set of topics, which, in the context of their role as members of such a set, are called ‘themes’. Each theme contributes to the extent of the scope that the themes collectively define; a given scope is the union of the subjects of the set of themes used to specify that scope.

NOTE 3 If it is desired to specify a scope which is the intersection (rather than the union) of two topics, this can be accomplished by creating a topic whose subject is that intersection, and then by using that topic as a theme.

3.17 sort name

- a) A subelement (*sortname*) of a *topname* subelement of a topic link, containing a string that is an alternative representation of a topic name that is intended to be used for alphabetic or other ordering.
- b) A name characteristic of a topic that is specified in the content of a *sortname* element.

3.18 subject

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.

NOTE 4 The invisible heart of every topic link is the subject that its author had in mind when it was created. In some sense, a topic link reifies a subject. The *identity* attribute of a topic link is provided to allow the author of the topic link to indicate, as unambiguously as possible, the subject he had in mind as the organizing principle of the topic. See the definition of ‘subject descriptor’.

3.19 subject descriptor

Information which is intended to provide a positive, unambiguous indication of the identity of a subject, and which is the referent of an *identity* attribute of a topic link. (See also the definition of ‘public subject descriptor’.)

NOTE 5 There is no requirement that a subject descriptor be text, although it can be the text of a definition of the subject. It can also, for example, be a listing in a catalog of subjects, such as an acquisition number of an asset in a museum collection, a catalog number in a sales catalog, or a subject heading in a catalog of library subject headings. The distinction between a subject descriptor that happens to be a definition and an ordinary occurrence of a definition is that, in the case of the subject descriptor, the topic link’s author has indicated (by referring to it by means of the value of the *identity* attribute) that it is to be regarded as the authoritative definition of the organizing principle of the topic link. In the other case, by characterizing a definition as a definitional occurrence, the author is merely acknowledging the existence of the definition and its possible relevance to the subject of the topic link.

NOTE 6 Subject descriptors may be offline resources.

3.20 theme

A member of the set of topics comprising a scope within which a topic characteristic assignment is valid. See also the definitions of ‘scope’ and ‘topic’.

3.21 topic

- a) An aggregate of topic characteristics, including zero or more names, occurrences, and roles played in associations with other topics, whose organizing principle is a single subject.
- b) A topic link element. See 5.2.

3.22 topic association

- a) A specific relationship among specific topics that is asserted by an association link element.
- b) An association link element. See 5.3.

3.23 topic characteristic

Any defining characteristic of a topic. There are three kinds of topic characteristics:

- a) names,
- b) occurrences, and
- c) roles played in relationships ('associations') with other topics.

For example, a name of a topic is a 'name characteristic' of that topic.

3.24 topic characteristic assignment

- a) The mechanism whereby a topic characteristic becomes a characteristic of a topic. For example, *topname* subelements of topic link elements are used to assign names to topics as topic characteristics, so, in topic map documents, they perform the function of assigning topic name characteristics.
- b) The fact that a particular topic characteristic is a characteristic of a particular topic.

3.25 topic link

A hyperlink element conforming to the topic link architectural form defined in 5.2 by this International Standard.

NOTE 7 In this International Standard, the foregoing definition is invoked by the phrase 'topic link', or, since the default SGML name of the topic link architectural form is 'topic', by the special typography used to distinguish SGML names (i.e., *topic*).

NOTE 8 See also the definition of 'topic'.

3.26 topic map

- a) A set of information resources regarded by a topic map application as a bounded object set whose hub document is a topic map document conforming to the SGML architecture defined by this International Standard.
- b) Any topic map document conforming to the SGML architecture defined by this International Standard, or the document element (*topicmap*) of such a document.

- c) The document element type (*topicmap*) of the topic map document architecture.

3.27 topic name

- a) A string of characters specified as a name of a topic; a name characteristic of a topic.
- b) A topic name (*topname*) element, as defined by this International Standard.
- c) Either a base name (*basename*), display name (*dispname*) or name to be used as sort key (*sortname*) element, as defined by this International Standard, and/or the information that such an element contains.
- d) A combination of the foregoing definitions.

3.28 topic occurrence

Information that is specified as relevant to a given subject.

NOTE 9 Topic occurrences may be offline resources.

3.29 topic type

- a) A subject which is a class of topics.
- b) One of the classes of topics of which a particular topic link is an instance. The topic types of which a given topic link is an instance can be specified via its optional *types* attribute.

3.30 unconstrained scope

The scope comprised of all of the topics in a topic map. When no applicable *scope* attributes are explicitly specified as governing a topic characteristic assignment, the scope within which the topic characteristic assignment is made is the unconstrained scope.

NOTE 10 In other words, the unconstrained scope is the default scope. Thus, for example, in a given topic map, if no *scope* attributes are explicitly specified for the name characteristics of any topics, any two topic links that have any of the same names will be merged, due to the effect of the topic naming constraint.

4 Notation

Topic Maps is an enabling document architecture whose definition (this International Standard) conforms to the Architectural Form Definition Requirements in Normative Annex A.3 of ISO/IEC 10744:1997, the *SGML Architectural Form Definition Requirements (AFDR)*. The formal definition of the topic map notation is expressed as a meta-DTD. The specification of the Topic Maps architecture is accomplished by a combination of narrative text and formal definitions.

Any references in this document to industry and proprietary standards, products, user groups, and publications are not normative, and do not imply endorsement by ISO, IEC, or their national member bodies or affiliates. Any brand names or trademarks mentioned are the property of their respective owners.

The formal definitions are expressed in SGML.

The formal SGML definitions are part of the text of this International Standard and are protected by copyright. In order to facilitate conformance to the Topic Maps architecture, the formal SGML definitions may be copied as specified in the following copyright notice: *Copyright (C) 1999 International Organization for Standardization. Permission to copy in any form is granted for use with conforming Topic Maps systems and applications as defined in ISO/IEC 13250:1999, provided this notice is included in all copies.* The permission to copy does not apply to any other material in this International Standard.

NOTE 11 This document uses editorial conventions mandated by the ISO with which the reader should be familiar in order to understand the implications of certain words.

The text describing each construct emphasizes semantics, while the formal SGML definition provides the rigorous syntactic definitions underlying the text descriptions.

NOTE 12 For this reason, it is recommended that the reader refer to the SGML definitions while reading the textual descriptions. Although the SGML definition always follows the related text, the user may find it helpful to read the SGML first in some cases. The meta-DTD is found in "Annex A: Topic Maps Meta-DTD (Normative)".

When a construct is first introduced, it is described in the text. If the construct occurs in the formal SGML specification, both the formal SGML name and a full name in English are presented, as follows:

— The element form **full construct name** (*SGMLname*) ...

— The attribute **full construct name** (*SGMLname*) ...

The declarations include comments, called "conventional comments", that follow conventions established in the HyTime standard to specify syntactic and semantic constraints and other information that is known to an architecture engine such as a Topic Maps engine. The conventional comments do not extend SGML in any way. They are used in the architecture definitions only, as a notation for the documentation of the architecture. They need not be included in application DTDs and, if they are included, the SGML parser will treat them as it would any other comment.

4.1 RCS name, full name, description, and clause

Every form name is followed by comments giving the full name, a description of the form, and the number of the clause in this International Standard in which the form is defined.

Individual attributes have a full name and description comment.

4.2 Constraints

Comments labeled "Constraints" define additional semantic or syntactic constraints on the constructs they follow. Constraint comments that follow the name of a form define constraints on the use of that form in general. Constraint comments that follow a component of a declaration (for example, the default value prescription of an attribute declaration), define the specific constraints on that component.

4.3 Note

Note comments provide additional information not provided by the other comment types and are informational rather than constraining.

5 Topic Maps architecture

This clause defines the syntax of topic maps. The Topic Maps syntax makes use of the base, location address, and hyperlinking modules of the HyTime architecture as defined in clauses 6, 7 and 8 of ISO/IEC 10744:1997.

NOTE 13 The entire formal definition of the Topic Maps syntax, the Topic Maps meta-DTD, is found in normative Annex A of this International Standard.

When interchanged, topic maps are HyTime bounded object sets (BOSs). The hub document of such a BOS must contain a Topic Maps architectural support declaration (see Annex B for examples).

Only one of the hyperlink syntaxes defined by HyTime is used in the topic map syntax: **variable link** (*varlink*).

The HyTime architecture provides a comprehensive set of addressing mechanisms and a standard syntax for using them. In addition, it provides means whereby any addressing syntax can be declared and used. The topic map architecture preserves these features of HyTime. Thus, the Topic Maps architecture allows topic map authors to use any addressing scheme, including proprietary addressing mechanisms driven by expressions in any notations, provided each such notation is formally declared as a notation in the manner prescribed by the SGML and HyTime International Standards.

NOTE 14 For example, in an XML environment, location addressing can be accomplished using IETF Uniform Resource Locator (URL) notation.

5.1 Topic Map Architectural Form

The **topic map** (*topicmap*) element form is used as the document element of all documents that conform to the Topic Maps architecture defined by this International Standard.

The effect of specifying the **added themes** (*addthemes*) attribute is to add the themes that it references to the scopes of all of the topic characteristic assignments made throughout the document of which the element is the root element.

NOTE 15 See the definition of 'added themes'.

NOTE 16 The *addthemes* attribute can be used to acknowledge and document the fact that the document specifies only topic characteristic assignments that are within the scope defined by the set of themes that it specifies. It can be used to avoid specifying these common themes explicitly in every scope. After a topic map document is merged with other topic map documents, the contributions that it made to the resulting merged topic map can be distinguished from the contributions of all others by virtue of the fact that everything it contributed continues to appear within the scopes of the topics specified by the *addthemes* attribute of its document element.

The *topicmap* element type is derived from the document element type of the HyTime architecture (*HyDoc*). All of the remaining attributes (*maxbos*, *boslevel* and *grovplan*) are inherited from *HyDoc*. The optional *maxbos* and *boslevel* attributes are used in hub documents in specifying the members of the HyTime bounded object set rooted at the document. The optional *grovplan* attribute is used in HyTime addressing. (See ISO/IEC 10744:1997.)

NOTE 17 As the use of the TMCFC parameter entity indicates, valid topic map documents may or may not have any topic links, association links or facet links in them. Some conforming applications may support only *facet* element types, while others may not support *facet* element types.

```

<!entity %
  TMCFC          -- Topic map context-free content --
  "topic|assoc|facet|bosspec|addthms|TMBrid"
>
<!element
  TMBrid          -- Topic map bridge element --
  - O
  ANY
>
<!element
  topicmap        -- Topic map document element --
                  -- Clause: 5.1 --
  - O
  (%TMCFC;)*
>
<!attlist
  topicmap
  HyTime          -- HyTime architectural form name --
    NAME
  HyDoc           -- HyTime document element. (This
                  attribute definition is redundant; it
                  appears here as an aid to
                  understanding.) --
  addthms         -- Added themes --
                  -- Themes to add to all scopes that govern
                  the assignments of topic names,
                  occurrences, and roles played in
                  associations in this topic map
                  document. --
    CDATA         -- Reference --
                  -- Reftype: topic+ --
    #IMPLIED      -- Default: No themes added via this
                  attribute. --
  -- bos --       -- HyTime bounded object set --
                  -- HyTime Clause: 6.5.1 --
  maxbos          -- Maximum bounded object set level --
                  -- Bounding level of HyTime bounded object
                  set when document is a hub or
                  subhub. --
    NUMBER        -- Constraint: Depth of nested entities to
                  include in BOS (0=no limit, 1=hub only)
                  --
    0
  boslevel        -- Bounded object set level --
                  -- Default BOS level used by data entities
                  declared in hub document. --
    NUMBER        -- Constraint: Depth of nested entities to
                  include in BOS (0=no limit, 1=this
                  entity only) --
    #IMPLIED      -- Default: No HyTime BOS --
  -- bosspecat -- -- BOS exception specification attributes
                  --
                  -- HyTime Clause: 6.5.3 --
  bosspec         -- Bounded object set exception
                  specification --

```

```

-- Adjustments to be made to the bounded
-- object set. --
IDREFS -- Reference --
-- Reftype: bosspec+ --
-- Constraint: Must be internal reference
--
#IMPLIED -- Default: No BOS exception specification
--
-- dgrvplan -- -- HyTime document grove plan --
-- HyTime Clause: 7.1.4.1 --
grovplan -- Grove plan --
-- Grove plan for HyTime extended SGML
-- document grove --
CDATA -- Reference --
-- Reftype: grovplan --
#IMPLIED -- Default: HyTime default grove plan --
>

```

5.2 Topic link

5.2.1 Topic Link Architectural Form

The **topic link** (*topic*) element form is used to assign topic name characteristics and topic occurrence characteristics to a topic.

Every topic link is intended by its author to be organized around exactly one subject, regardless of whether that subject is explicitly defined anywhere. A topic link may declare zero or more names and zero or more pieces of information ('occurrences') that are relevant to its subject. Names, and the scopes within which the names are applicable to the subject, are declared by means of *topname* subelements. Occurrences are the anchors of the topic link; these, and the scopes within which the occurrences are applicable to the subject, are specified by means of *occurs* subelements.

The required **unique identifier** (*id*) attribute facilitates the addressing of topics by association links, by the *identity* attributes of other topic links, and, in their roles as themes in scopes, by *scope* and *addthems* attributes.

The optional **subject identity** (*identity*) attribute refers to one or more indications ('subject descriptors') of the identity of the subject (the organizing principle) of the topic link. All of the other topic characteristics specified by the topic link are regarded as elaborating, and in no way contradicting, the subject described by the subject descriptor(s), if any. There are no restrictions on the kinds of information that may be referenced by an *identity* attribute.

NOTE 18 The information referenced by an *identity* attribute may or may not take the form of a topic link in a topic map document, may or may not be text, may or may not be machine-interpretable, and may or may not be online.

Any two or more topic links that reference the same subject by means of their *identity* attributes are equivalent to a single topic link that has the union of the characteristics (the names, occurrences, and associations) of both topic links. The two or more topic links may be merged, and/or applications may process and/or render them as if they have been merged.

NOTE 19 The two or more topic links do not have to refer to the same subject descriptor in order to be merged under this rule. It is only necessary that the subject that is somehow indicated by the two *identity* attributes be one and the same subject. If two or more topics refer to exactly the same subject descriptor, the subject descriptor may be described as a

‘public subject descriptor’, and it becomes possible to automate the merging of all such topics by making the assumption that, if they all share the same subject descriptor, they all share the same subject identity.

Similarly, if the *identity* attribute references one or more topic links, topic map processing applications must regard the referencing topic link, and all the referenced topic links, as having one and the same subject, and therefore they may all be merged.

The optional **topic types** (*types*) attribute references one or more topic links. The subject of each such referenced topic link is a class of subject of which the subject of the referencing topic link is an instance. The class-instance relationship established between the subject of each referenced topic link and the subject of the referencing topic link could alternatively be established by a topic association link whose semantic is the relationship between a class and an instance of that class.

NOTE 20 In other words, the *types* attribute establishes a relationship between topics (a topic association), rather than being a means whereby the referencing topic becomes an occurrence of each of the referenced topics.

The topic relationships established by the *types* attribute are not superclass-subclass relationships. They are only class-instance relationships.

NOTE 21 Superclass-subclass relationships between topics can be asserted by topic association links that have been user-defined for that purpose.

The optional **scope** (*scope*) attribute references the themes that are added to the scopes within which all names and occurrences specified by the topic link are valid.

NOTE 22 The *scope* attribute of the topic link architectural form is designed to permit a reduction in syntactic redundancy by providing a means whereby the themes that are common to the scopes within which all the names and occurrences of a topic are valid can be specified once for all. There is no requirement that it be used, however, even if its use would reduce redundancy.

A valid topic link must have at least one of the following: a topic name, a topic occurrence, or a role played in an association with at least one other valid topic.

The *topic* element type is derived from the *varlink* element type of the HyTime architecture.

NOTE 23 However, because it is possible to have a topic link without any *occurs* elements inside it, while (at the time of publication of this International Standard) the HyTime architecture requires that *varlink* elements always contain at least one *anchspec*, it is necessary that the value of the HyTime attribute be *HyBrid* when there are no contained *occurs* elements, and that it be *varlink* when there are any such contained *occurs* elements. It is anticipated that future versions of the HyTime architecture will permit *varlink* elements that contain no *anchspec* elements, and if such a change in the HyTime standard is made, it will be possible to fix the value of the *HyTime* attribute of all *topic* elements at *varlink*.

The optional **hyperlink type** (*linktype*) attribute is defined by the HyTime architecture in order to allow the link type name to be different from the generic identifier.

NOTE 24 Neither the value of the *linktype* attribute nor the generic identifier of a topic link has any significance with respect to the topic mapping semantics defined by this International Standard.

```
<!element
  topic          -- Topic link --
                -- Clause: 5.2.1 --
  - 0
  ( topname | occurs)*
>
<!attlist
  topic
```

```

HyTime      -- HyTime architectural form name --
(varlink|HyBrid)
varlink     -- Constraint: varlink must be specified
            when occurrences exist. If topic has no
            occurrences, it must be declared as a
            HyTime bridge element (HyBrid). --
id          -- Unique identifier --
  ID
  #REQUIRED
identity    -- Subject identity --
            -- Reference to information (one or more
            subject descriptors) that confers
            understanding of the identity of the
            subject of this topic link. --
            CDATA      -- Reference --
            #IMPLIED  -- Default: No subject descriptors; the
            subject must be inferred from the
            topic's characteristics. --
types       -- Topic types --
            -- Topics whose subjects are the classes
            of topics of which this topic is an
            instance. --
            CDATA      -- Reference --
            #IMPLIED  -- Reftype: topic+ --
            -- Default: No class-instance topic
            associations are established via this
            attribute. --
            -- Note: Some might still be specified by
            topic association links, however. --
scope       -- Scope --
            -- The themes that are added to the scopes
            of all the names and occurrences
            specified by this topic link. --
            CDATA      -- Reference --
            #IMPLIED  -- Reftype: topic+ --
            -- Default: No themes are added by this
            attribute. --
linktype    -- Hyperlink type --
  NAME
  #IMPLIED  -- Default: Generic identifier --
>

```

5.2.2 Topic Name Architectural Form

A topic may have zero or more name characteristics (topic names). Topic names are specified using **topic name** (*topname*) elements; all such names become topic characteristics of the topic whose subject is the subject of the containing topic link.

This International Standard distinguishes three kinds of topic name: **base name** (*basename*), **display name** (*dispname*), and **name used as sort key** (*sortname*), specified by means of the three corresponding element types that a *topname* element may contain.

The **scope** (*scope*) attribute of the *topname* element specifies the themes that are common to the scopes of all of the topic name characteristics specified by the contained *basename*, *dispname* and *sortname* elements. The scope is the context (or area of validity) in which the name characteristic(s) specified by a *topname* element is/are assigned to the topic whose subject is the subject of the containing topic link.

The *scope* attributes of the contained name elements (*basename*, *dispname* and *sortname*) may be used to add more themes on a name-by-name basis, in the same manner as the *scope* attribute of the containing *topname* element.

NOTE 25 Thus, the *scope* attribute of the *topname* element form is really just a means of avoiding the syntactic redundancy of specifying the themes common to the contained elements separately via the *scope* attribute of each contained element.

NOTE 26 See also the definitions of ‘theme’ and ‘scope’ in 3.

If no *scope* attribute is specified by a *basename*, *dispname* or *sortname* element and no *scope* attribute is specified by its containing *topname* nor by the containing *topname*'s containing *topic* link, then the scope of the name characteristic specified by that *basename*, *dispname* or *sortname* is unconstrained. If any of the aforementioned *scope* attributes are specified, then the scope is constrained to the themes specified by those scope attributes, even if the *scope* attributes specify no themes, plus any themes added via any applicable *addthms* elements in the bounded object set, plus any themes added via the *addthms* attribute of the containing *topicmap* document element.

The content of the optional **display name** (*dispname*) element specifies a name that is designed to be displayed by an application to a user, when the name specified by the *basename* elements within the same containing *topname* should not be used for display purposes.

NOTE 27 The display name can be used to specify an abbreviated name for use in situations where display resources are limited, or it can be a graphic expressed in some data content notation.

The content of the optional **name to be used as sort key** (*sortname*) element specifies a name that is designed to be used to represent the topic in a sorting process that arranges a list of topics in some order, when the name specified by the *basename* elements within the same containing *topname* should not be used for that purpose.

NOTE 28 Thus, the *basename* elements, at least one of which is required, is also, in effect, the default content of the optional *dispname* and *sortname* elements. If no *dispname* elements are specified, the *basename* elements are to be used as display names. Similarly, if no *sortname* elements are specified, the *basename* elements are to be used as sort keys.

The data content of both *basename* and *sortname* elements must be text strings, and they may be words or phrases. The data content of a *dispname* element may be either a text string or notation data; if it is notation data, it may be a displayable graphic or other information intended to identify the subject to one or more of the senses of the user of the topic map. If the content of a *dispname* is notation data, it must declare the notation via the HyTime-defined *notation* common attribute (see ISO/IEC 10744:1997).

NOTE 29 There are two reasons why base names, display names, and names used as sort keys may share a single containing *topname* element:

- a) to allow them to share a common scope, specified via the *scope* attribute of the containing *topname* element, and/or
- b) to indicate that base names, display names, and sort keys correspond to one another. Therefore, if a one-to-one relationship is desired between base names and display names, for example, each pair, consisting of one base name and one display name, must be contained in a separate *topname* element. However, if no display names or sort names are used, and many base names of a single topic have the same scope, all of the base names may appear within a single *topname* element.

This International Standard does not permit two distinct subjects to have the same name characteristic within exactly the same scope (the ‘topic naming constraint’). When topic maps are processed, each distinct set of themes that serves as a scope constitutes a namespace in which no two subjects can have the same name. If a conforming topic map application detects a situation in which multiple topic links have the same name characteristic within the same scope, they shall be merged.

NOTE 30 This means that applications that render the topic map will behave as though there was only a single topic link whose characteristics comprise the union of the topic characteristics of all of the topic links that had the same name within the same scope.

NOTE 31 The topic naming constraint is designed to preserve the ability to identify subjects unambiguously in terms of their topic name characteristics. The topic naming constraint is also necessary in order to support the most basic functionality of indexes and glossaries, which must make distinctions between the various meanings of words and phrases in order to support navigation to the relevant occurrences. Topic map authors must use scopes to distinguish between the different meanings of any name that is used for more than one subject. Consequently, if a topic map author does not wish to specify *scope* attributes explicitly, that author cannot use the same name for any two different subjects, because the default scope is a single scope: the unconstrained scope. Since any two identical names will appear within that single scope, the two subjects of which the two names are topic characteristics will be automatically merged. Such merging is erroneous and extremely undesirable unless the two topic links that have the same name in the same scope also have the same subject identity.

NOTE 32 As an aid to topic map authors, topic map authoring and merging applications may be designed in such a way as to give warning when topic links are being merged on account of the fact that they have the same name in the same scope.

NOTE 33 One of the effects of the topic naming constraint is that the merging of topic maps can be accomplished in such a way as to make the merger maintainable even when the member topic maps are maintained separately, asynchronously, and with no extra, agreed-upon discipline (such as some sort of element naming discipline) designed to permit easy maintenance of references among the component documents of the merged topic map. The topic naming constraint makes the addressing of the subjects covered by topic maps dependent only on their names and the distinguishing criteria of their names (the scopes within which their name characteristics are valid), and not on the organization or tagging of their corresponding topic links. The process of merging topic maps can be accomplished by creating a topic map hub document that specifies other topic maps as being members of its BOS, and that contains topic links whose names are the same as the names of topics in the member topic map documents, within the same scopes. The process of replacing one of the component topic maps with a newer version of itself is a matter of replacing the old version with the new version, and reprocessing the bounded object set in the usual way. There may be name and scope changes in the new version which will necessitate some thought and effort on the part of the operator in order to obtain the best possible merger, but the required effort will be limited to the resolution of issues created by the changes that were made to the component topic maps since the last merger was completed.

NOTE 34 If any two topic maps that are to be merged conflict with one another because they happen to provide the same name within the same scope for two different subjects, the merger of the different subjects can be prevented by applying different added themes to one or both of their containing topic map documents, using one or more *addthms* elements. The added themes specified by such *addthms* elements can serve to distinguish the two identical names, because they will no longer appear within exactly the same scope.

```
<!element
  topname          -- Topic name --
                  -- Clause 5.2.2--
  O O
  (basename+, dispname*, sortname* )
                  -- If dispnames or sortnames are not
                  -- specified, applications use basenames
                  -- for display and sorting purposes. --
>
<!attlist
  topname
  scope           -- Scope --
                  -- Reference to a set of themes (topic
                  -- links) to be added to the scopes of the
                  -- name characteristics specified by the
                  -- contained basename, dispname, and
                  -- sortname elements. Scopes are sets of
                  -- themes that collectively define the
                  -- limited context within which
                  -- characteristics are validly applicable
                  -- to the topic. --
```

```

        CDATA          -- Reference --
                        -- Reftype: topic+ --
        #IMPLIED      -- Default: No themes are added via this
                        attribute. --
>
<!element
  (basename | sortname)
                        -- Base name --
                        -- and --
                        -- Name to be used as sort key --
  - 0
  (#PCDATA)          -- String to be used as name --
>
<!element
  dispname          -- Display name --
  - 0
  (#PCDATA|TMbrid)*
                        -- String (or notation data) to be
                        displayed as name --
>
<!attlist ( basename | sortname | dispname)
  scope            -- Scope --
                        -- References to a set of themes (topic
                        links) to be added to the scope of the
                        name characteristic specified in the
                        content. --
        CDATA          -- Reference --
                        -- Reftype: topic+ --
        #IMPLIED      -- Default: No themes are added via this
                        attribute. --
>

```

5.2.3 Topic Occurrence Architectural Form

By means of location addresses specified in its content, the **topic occurrence** (*occurs*) element references information (one or more ‘occurrences’) that is relevant to the subject of the containing topic link. This International Standard imposes no constraints on the nature of information objects that can be specified as occurrences of topics, nor on the addressing notations used to reference such occurrences.

NOTE 35 Applications may impose such constraints.

NOTE 36 Topic occurrences may be offline resources.

All of the occurrences specified by any given *occurs* element fall into a single user-defined category of occurrences called an ‘occurrence role’ -- the role played by the occurrences in contributing to the information that participates in the subject characterized by the containing topic link. Within a single topic link, more than one *occurs* element may reference the same information, in which case the information plays multiple occurrence roles.

NOTE 37 For example, if the subject of a topic link is Leonardo Da Vinci, there may be a ‘scientific-biography’ occurrence role, and a separate ‘artistic-biography’ occurrence role. Some information resources may fall into both categories; if so, such resources will be referenced by both of the *occurs* elements that correspond to the two occurrence roles.

The occurrence role is the scope within which the occurrences are relevant to the subject of the containing topic link. The set of themes (topics) that constitute the scope is specified via the optional **scope** (*scope*) attribute. If no *scope* attribute is specified by an *occurs* element, and no *scope* attribute is specified by its containing *topic* link element, then the scope of the occurrence

characteristics specified by that *occurs* element is unconstrained. If either of the aforementioned *scope* attributes are specified, then the scope is constrained to the themes specified by those scope attributes (even if the *scope* attributes specify no themes), plus any themes added via any applicable *addthms* elements in the bounded object set, plus any themes added via the *addthms* attribute of the containing *topicmap* document element.

The optional *occr1* attribute can be used to provide a mnemonic name for the occurrence role. If the *occr1* attribute is not specified, the generic identifier is regarded as the mnemonic name of the occurrence role.

The optional **occurrence role type** (*type*) attribute references a single topic link. The subject of the referenced topic link is a class of occurrence role of which the occurrence role expressed by the *occurs* element is an instance. The class-instance relationship established between the subject of the referenced topic link and the referencing *occurs* element could alternatively be established by making the *occurs* element an occurrence of the referenced topic link, within the scope of an occurrence role whose meaning is that the occurrence role is an instance of the subject of the topic link.

If the **occurrence role type** (*type*) attribute is specified, and if the topic referenced by the *type* attribute has a name characteristic that lies within a scope that is appropriate to the topic map user's context, the referenced topic's name characteristic is used to characterize the occurrence role for the user. Otherwise, the value of the *occr1* attribute (or, if the *occr1* attribute is not specified, the generic identifier) is used to characterize the occurrence role for the user.

NOTE 38 The topic referenced via the *type* attribute can have many names in scopes designed for many different user contexts, including many different natural languages and delivery platforms, while the *occr1* attribute or generic identifier is just a single token. Therefore, the use of a topic, referenced by the *type* attribute, to characterize the occurrence role offers far more flexibility and representational power than the simple mnemonic naming facility offered by the *occr1* attribute or generic identifier.

The *occurs* element type is derived from the *anchspec* element type of the HyTime architecture. Most of the remaining attributes (*linktrav*, *listtrav*, *multmem* and *emptyanch*) are inherited from the *anchspec* element type defined by the HyTime standard; these have been given default values that are generally appropriate for topic map applications, but they are not necessarily the same as the default values specified by the HyTime architecture (see ISO/IEC 10744:1997). The *HyNames* architectural control attribute is used here to indicate that for HyTime processing purposes the *occr1* attribute should be regarded as the HyTime *anchrole* attribute. (See ISO/IEC 10744:1997.)

```
<!element
  occurs          -- Topic occurrence --
                 -- Clause: 5.2.3 --
  - O
  (%loc;)*
>
<!attlist
  occurs
  HyTime         -- HyTime architectural form name --
  NAME
  #FIXED
  anchspec
  scope          -- Scope --
                 -- Reference to themes that are added to
                 the scope within which the occurrences
                 are applicable to the topic
                 characterized by the containing topic
                 link.--
  CDATA         -- Reference --
```

```

#IMPLIED -- Reftype: topic+ --
-- Default: No themes are added to the scope
-- by means of this attribute. --
occr1 -- Occurrence role name --
-- Note: Not displayed for the topic map
-- user if the topic referenced by the
-- type attribute has displayable
-- characteristics within the user's
-- scope. --
NAME
#IMPLIED -- Default: GI of element is treated as
-- occurrence role name. --
type -- Occurrence role type --
-- Reference to the topic that names
-- and/or otherwise characterizes the
-- occurrence role. The characteristics
-- of the referenced topic, if
-- appropriate, will be displayed to the
-- user instead of the value of the occr1
-- attribute. --
CDATA -- Reference --
#IMPLIED -- Reftype: topic --
-- Default: No topic characterizes the
-- occurrence role, unless this element is
-- an occurrence (with an occurrence role
-- whose meaning is instance) of a topic
-- whose subject is the nature of the
-- occurrence role. The value of the
-- occr1 attribute will be displayed as
-- the occurrence role name. --
linktrav -- Hyperlink traversal rules --
-- Traversal between anchors of hyperlinks:
-- A any traversal or departure (EID)
-- D departure after internal arrival
-- E traversal after external arrival
-- I traversal after internal arrival
-- N no traversal after internal arrival
-- P no internal arrival
-- R return traversal after internal arrival --
NAMES -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|"
-- "I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
A
listtrav -- List traversal rules --
-- Traversal between members of list anchors:
-- A adjacent (both left and right) traversal
-- L left traversal
-- N no traversal
-- R right traversal
-- W wrapping traversal --
NAMES -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
N -- Default: Show the whole list --
multmem
(single|list|corlist)
list
emptyanc
(error|noterror)
error
HyNames
CDATA
"anchrole occr1"
>

```

5.3 Association Link

5.3.1 Association Link Architectural Form

The **association link** (*assoc*) element form is used to express relationships among topics. Topic Maps applications define the nature of the relationships, and of the roles played by topics in those relationships.

The optional **scope** (*scope*) attribute specifies the scope (the set of themes) within which the association is applicable to the topics that serve as anchors of the association link. If the *scope* attribute is not specified, the scope is unconstrained. If the *scope* attribute is specified, then the scope is constrained to the themes specified by the *scope* attribute (even if the *scope* attribute specifies no themes), plus any themes added via any applicable *addthms* elements in the bounded object set, plus any themes added via the *addthems* attribute of the containing *topicmap* document element.

The optional **hyperlink type** (*linktype*) attribute can be used to provide a mnemonic name for the association type. If the *linktype* attribute is not specified, the generic identifier is regarded as the mnemonic name of the association type.

The optional **association type** (*type*) attribute references a single topic link. The subject of the referenced topic link is a class of association of which the association expressed by the association link is an instance. The class-instance relationship established between the subject of the referenced topic link and the referencing association link could alternatively be established by making the association link an occurrence of the referenced topic link, with an occurrence role whose meaning is that the association link is an instance of the subject of the topic link.

NOTE 39 If topic links whose subjects are classes of topic associations specify *identity* attributes, and if the subject descriptor(s) referenced by the *identity* attributes describe the same subject, the association links that are instances of those classes can be universally recognized as assertions of equivalent relationships. Depending on the nature of such relationships, the use of public subject descriptors to define association types may significantly facilitate the process of merging topic maps, even when they emanate from disparate sources.

If the **association type** (*type*) attribute is specified, and if the topic referenced by the *type* attribute has a name characteristic that lies within a scope that is appropriate to the topic map user's context, the referenced topic's name characteristic is used to characterize the association type for the user. Otherwise, the value of the *linktype* attribute (or, if the *linktype* attribute is not specified, the generic identifier) is used to characterize the association type for the user.

NOTE 40 The topic referenced via the *type* attribute can have many names in scopes designed for many different user contexts, including many different natural languages and delivery platforms, while the *linktype* attribute or generic identifier is just a single token. Therefore, the use of a topic, referenced by the *type* attribute, to characterize the association type offers far more flexibility and representational power than the simple mnemonic naming facility offered by the *linktype* attribute or generic identifier.

```
<!element assoc      -- Association link --
  -- Clause: 5.3.1 --
  - O (assocr1)+ >
<!attlist assoc
  HyTime             -- HyTime architectural form name --
  NAME
  #FIXED
  varlink
  scope              -- Scope --
```



```

-- Reference to themes that are added to
-- the scope within which the association
-- is applicable. --
    CDATA          -- Reference --
-- Reftype: topic+ --
    #IMPLIED      -- Default: Scope is unconstrained. --
linktype         -- Hyperlink type. --
-- Mnemonic name for the association
-- type. --
-- Note: Not displayed for the topic map
-- user if the topic referenced by the
-- type attribute has displayable
-- characteristics within the user's
-- scope. --
    NAME
    #IMPLIED      -- Default: Generic identifier --
type            -- Association type --
-- Topic whose subject is the class of
-- association of which this association
-- is an instance. --
    CDATA          -- Reference --
-- Reftype: topic --
    #IMPLIED      -- Default: No type is specified by this
-- attribute. --
-- Note: A type might exist by virtue of
-- the fact that this association link is
-- an occurrence (where the occurrence
-- role means "instance") of a topic whose
-- subject is the nature of the
-- association, however. --
>

```

5.3.2 Association Role Architectural Form

The **association role** (*assocrl*) element form specifies a user-defined role played by one or more specific topics in the relationship asserted by the containing association link element. The topics that play the role, if any, are referenced by means of the location addresses specified in the content of the association role element. Within a single association link, more than one *assocrl* element may reference the same topic, in which case the topic plays multiple roles in the association.

NOTE 41 Thus, the containing *assoc* element can assert that a topic has one or more specific relationships to itself.

Regardless of the association role(s) they play in the relationship expressed by the containing *assoc* element, all topics referenced in the content of the contained *assocrl* elements play their roles in that relationship within the same scope.

NOTE 42 This is the reason why there is no *scope* attribute on the *assocrl* element form.

The optional HyTime-defined **anchor role** (*anchrole*) attribute can be used to provide a mnemonic name for the association role. If the *anchrole* attribute is not specified, the generic identifier is regarded as the mnemonic name of the association role.

The optional **association role type** (*type*) attribute references a single topic link. The subject of the referenced topic link is a class of association role of which the association role expressed by the *assocrl* element is an instance. The class-instance relationship thus established between the subject of the referenced topic link and the referencing *assocrl* element could alternatively be established by making the *assocrl* element an occurrence of the referenced topic link, within the scope of an

occurrence role whose meaning is that the association role is an instance of the subject of the topic link.

NOTE 43 If the topic links whose subjects are association role types specify *identity* attributes, and if the subject descriptor(s) referenced by those *identity* attributes describe the same subject, the *assocrl* elements that are instances of those association role types can be universally recognized as specifications of equivalent association roles. Depending on the nature of such association roles, the use of public subject descriptors to define association role types may significantly facilitate the process of merging topic maps, even when they emanate from disparate sources.

If the **association role type** (*type*) attribute is specified, and if the topic referenced by the *type* attribute has a name characteristic that lies within a scope that is appropriate to the topic map user's context, the referenced topic's name characteristic is used to characterize the association role for the user. Otherwise, the value of the *anchrole* attribute (or, if the *anchrole* attribute is not specified, the generic identifier) is used to characterize the association role for the user.

NOTE 44 The topic referenced via the *type* attribute can have many names in scopes designed for many different user contexts, including many different natural languages and delivery platforms, while the *anchrole* attribute or generic identifier is just a single token. Therefore, the use of a topic, referenced by the *type* attribute, to characterize the association role offers far more flexibility and representational power than the simple mnemonic naming facility offered by the *anchrole* attribute or generic identifier.

The *assocrl* element type is derived from the *anchspec* element type of the HyTime architecture. The remaining attributes (*linktrav*, *listtrav*, *multmem* and *emptyanch*) are inherited from the *anchspec* element type defined by the HyTime standard; these have been given default values that are generally appropriate for Topic Maps applications, but which are not necessarily the default values specified by the HyTime architecture (see ISO/IEC 10744:1997).

```

<!element
  assocrl          -- Association role --
                  -- Clause: 5.3.2 --
  - 0
  (%loc;)+        -- Reftype: topic+ --
>
<!attlist
  assocrl
  HyTime          -- HyTime architectural form name --
  NAME
  #FIXED
  anchspec
  anchrole        -- Anchor role --
                  -- Note: Not displayed for the topic map
                  -- user if the topic referenced by the
                  -- type attribute has displayable
                  -- characteristics within the user's
                  -- scope. --
  NAME
  #IMPLIED        -- Default: GI of element is treated as
                  -- anchor role. --
  type            -- Association role type --
                  -- Reference to the topic that names
                  -- and/or otherwise characterizes the
                  -- association role. The characteristics
                  -- of the referenced topic, if
                  -- appropriate, will be displayed to the
                  -- user instead of the value of the
                  -- anchrole attribute. --
  CDATA          -- Reference --
                  -- Reftype: topic --
  #IMPLIED        -- Default: No topic characterizes the
                  -- association role, unless this element
                  -- is an occurrence (with an occurrence

```

```

role whose meaning is instance) of a
topic whose subject is the nature of
the association role. The value of the
anchrole attribute will be displayed as
the association role name. --
linktrav -- Hyperlink traversal rules --
-- Traversal between anchors of hyperlinks:
A any traversal or departure (EID)
D departure after internal arrival
E traversal after external arrival
I traversal after internal arrival
N no traversal after internal arrival
P no internal arrival
R return traversal after internal arrival --
NAMES -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|
"I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
A
listtrav -- List traversal rules --
-- Traversal between members of list anchors:
A adjacent (both left and right) traversal
L left traversal
N no traversal
R right traversal
W wrapping traversal --
NAMES -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
N -- Default: Show the whole list --
multmem
(single|list|corlist)
list
emptyanc
(error|noterror)
error
>

```

5.4 Themes To Be Added Architectural Form

The **themes to be added** (*addthms*) element allows themes to be added:

- to all the scopes of all topic characteristic assignments (that is, all topic names, topic occurrences, and roles played in associations with other topics) specified by topic links and topic associations in the topic map documents referenced via the **topic map document entities** (*tmdocs*) attribute, and/or
- to the scopes of all topic names and topic occurrences specified by specific topic links (and the subelements of topic links) referenced via the **characteristic assigners** (*cassign*) attribute, if any, and/or
- to the scopes of all roles played by topics in topic associations specified by specific association links referenced via the **characteristic assigners** (*cassign*) attribute, if any.

The **added themes** (*addthems*) attribute's value is a reference to one or more topic link elements. The referenced topic link elements must be regarded by topic map applications as additional themes in the scopes of the topic characteristics specified via the *tmdocs* and *cassign* attributes.

The *tmdocs* and *cassign* attributes are independent of one another. If both are specified, the *tmdocs* attribute does not establish a location source for the addresses specified via the *cassign* attribute; the *cassign* attribute must be used in such a way that it establishes its own location source(s).

NOTE 45 When topic maps are to be merged, the *tmdocs* attribute can be used to allow applications to distinguish between the characteristics of topics in terms of the different topic maps that contributed those characteristics. For example, a topic can be created that represents the rhetorical position or purpose of a given topic map, and then, by means of an *addthms* element, that a new topic can be used as an additional scope within which all the topic characteristics specified by the topic map are said to be valid. After the topic map document is merged with other topic map documents, the contributions that it made to the resulting merged topic map can be distinguished from all others by virtue of the fact that everything it contributed continues to appear within the scope of the topic representing the document or hyperdocument that contributed it.

NOTE 46 The *addthms* element's content is not defined by the Topic Maps architecture.

```

<!element
  addthms          -- Themes to be added --
                  -- (To scopes specified by topic map
                  -- documents and/or by topic links and/or
                  -- association links.) --
                  -- Clause: 5.4 --
  - O
  (TMBrid)*       -- No content defined by the Topic Maps
                  -- architecture --
>
<!attlist
  addthms         -- Themes to be added --
                  -- Clause: 5.4 --
  addthems       -- Added themes --
                  -- Themes to be added to the scopes
                  -- specified by the tmdocs and cassign
                  -- attributes --
  CDATA          -- Reference --
                  -- Reftype: topic+ --
  #REQUIRED
  tmdocs         -- Topic map document entities --
  ENTITIES       -- Constraint: Must be one or
                  -- more document entities of
                  -- topic map documents. --
  #IMPLIED
  cassign        -- Characteristic assigners --
                  -- Elements that assign characteristics to
                  -- topics. The themes specified by the
                  -- addthms attribute are to be added to
                  -- the scopes within which the
                  -- characteristics they specify are
                  -- regarded as valid --
  CDATA          -- Reference --
                  -- Reftype: (topic | topname | basename |
                  -- dispname | sortname | occurs | assoc)+
                  --
  #IMPLIED
>

```

5.5 Facet Linking

By means of the *facet linking* facility, property/value pairs can be added to read-only information objects. The properties are called *facet types*, and the values are called *facet values*. This International Standard does not constrain the nature of facet linking applications; they may or may not also use topic links.

NOTE 47 The property/value pairs applied by facet links can be used, for example, as selection criteria to create partial views of a corpus of information.

NOTE 48 Topic links are much more generalized and powerful than facet links.

5.5.1 Facet Link Architectural Form

The **facet link** (*facet*) element form is used to apply property/value pairs to information objects specified by the contained *fvalue* elements. Facet link properties ('facet types') and values (specified by means of the contained *fvalue* elements) are user-defined.

The optional **hyperlink type** (*linktype*) attribute can be used to provide a mnemonic name for the property (facet type). If the *linktype* attribute is not specified, the generic identifier is regarded as the mnemonic name of the property.

The optional **facet type** (*type*) attribute references a single topic link. The subject of the referenced topic link is the property (the facet type) specified in all of the property/value pair assignments made by the facet link. The class-instance relationship established between the subject of the referenced topic link and the referencing facet link could alternatively be established by making the facet link an occurrence of the referenced topic link, with an occurrence role whose meaning is that the facet link is an instance of the subject of the topic link. If the **facet type** (*type*) attribute is specified, and if the topic referenced by the *type* attribute has a name characteristic that lies within a scope that is appropriate to the topic map user's context, the referenced topic's name characteristic is used to characterize the property (the facet type) for the user. Otherwise, the value of the **hyperlink type** (*linktype*) attribute (or, if the *linktype* attribute is not specified, the generic identifier) is used to characterize the property for the user.

NOTE 49 The topic referenced via the *type* attribute can have many names in scopes designed for many different user contexts, including many different natural languages and delivery platforms, while the *linktype* attribute or generic identifier is just a single token. Therefore, the use of a topic, referenced by the *type* attribute, to characterize the property (the facet type) offers far more flexibility and representational power than the simple mnemonic naming facility offered by the *linktype* attribute or generic identifier.

The *facet* element type is derived from the *varlink* element type of the HyTime architecture.

```
<!element
  facet                -- Facet link --
                      -- Clause: 5.5.1 --
  - 0
  (fvalue)+
>
<!attlist facet
  HyTime              -- HyTime architectural form name --
  NAME
  #FIXED
  varlink
  linktype            -- Hyperlink type. --
                      -- Mnemonic name for the property (facet
                      -- type). --
                      -- Note: Not displayed for the topic map
                      -- user if the topic referenced by the
                      -- type attribute has displayable
                      -- characteristics within the user's
                      -- scope. --
  NAME
  #IMPLIED           -- Default: Generic identifier --
  type                -- Facet type --
                      -- Topic whose subject is the property of
                      -- the property/value pair(s) being
                      -- assigned to the anchor(s). --
  CDATA               -- Reference --
```

```

#IMPLIED          -- Reftype: topic --
                  -- Default: No facet type topic is
                  -- specified by this attribute. --
                  -- Note: A facet type topic might exist by
                  -- virtue of the fact that this facet link
                  -- is an occurrence (where the occurrence
                  -- role means "instance") of a topic whose
                  -- subject is the nature of the property,
                  -- however. --
>

```

5.5.2 Facet Value Architectural Form

The **facet value** (*fvalue*) element form specifies a user-defined value of the property (facet type) being applied by the containing facet link. The information objects to which the property/value pair is being assigned are referenced by means of the location addresses specified in the content of the *fvalue* element.

The optional **facet value name** (*facetval*) attribute specifies the token which is the value of the property/value pair being assigned. If the *facetval* attribute is not specified, the generic identifier of the *fvalue* is the value being assigned.

The optional **facet value type** (*type*) attribute references a single topic link. The subject of the referenced topic link is the significance of the facet value. The class-instance relationship established between the subject of the referenced topic link and the referencing *fvalue* element could alternatively be established by making the *fvalue* element an occurrence of the referenced topic link, with an occurrence role whose meaning is that the *fvalue* element is an instance of the subject of the topic link.

The *fvalue* element type is derived from the *anchspec* element type of the HyTime architecture. The attributes (*linktrav*, *listtrav*, *multmem* and *emptyanch*) are inherited from the *anchspec* element type defined by the HyTime standard; these have been given default values that are generally appropriate for *fvalue* elements, but which may differ from the default values specified in the HyTime standard (see ISO/IEC 10744:1997). The *HyNames* architectural control attribute is used here to indicate that for HyTime processing purposes the *facetval* attribute should be regarded as the HyTime *anchrole* attribute. (See ISO/IEC 10744:1997.)

```

<!element
  fvalue          -- Facet value --
                 -- Clause: 5.5.2 --
  - 0
  (%loc;)*
>
<!attlist
  fvalue
    HyTime       -- HyTime architectural form name --
    NAME
    #FIXED
    anchspec
    facetval     -- Facet value name --
                 -- Token is value of property being
                 -- assigned. --
    NAME
    #IMPLIED     -- Default: Facet value name is GI of
                 -- element. --
    type         -- Facet value type --
                 -- Reference to a topic whose subject is
                 -- the significance of the facet value

```

```

        name. --
    CDATA      -- Reference --
              -- Reftype: topic --
    #IMPLIED  -- Default: No facet value type topic is
              -- specified by this attribute. --
              -- Note: A facet value type topic might
              -- exist by virtue of the fact that this
              -- fvalue element is an occurrence (where
              -- the occurrence role means "instance")
              -- of a topic whose subject is the
              -- significance of the facet value name,
              -- however. --
    linktrav  -- Hyperlink traversal rules --
              -- Traversal between anchors of hyperlinks:
              -- A any traversal or departure (EID)
              -- D departure after internal arrival
              -- E traversal after external arrival
              -- I traversal after internal arrival
              -- N no traversal after internal arrival
              -- P no internal arrival
              -- R return traversal after internal arrival --
    NAMES     -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|"
              -- "I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
    A
    listtrav  -- List traversal rules --
              -- Traversal between members of list anchors:
              -- A adjacent (both left and right) traversal
              -- L left traversal
              -- N no traversal
              -- R right traversal
              -- W wrapping traversal --
    NAMES     -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
    N         -- Default: Show the whole list --
    multmem   (single|list|corlist)
              list
    emptyanc  (error|noterror)
              noterror
    HyNames   CDATA
              "anchrole facetval"
>

```

6 Conformance

If a topic map document complies with all provisions of this International Standard, is a conforming SGML document as defined in ISO 8879, and is a conforming HyTime document as defined in ISO/IEC 10744, it is a conforming topic map document.

A conforming application intended to use existing topic maps must be able to:

- parse the interchange syntax,
- identify the topic map constructs defined by this International Standard, and
- apply whatever processing the application designers considered appropriate, in light of the requirements to be fulfilled by the application, and the semantics of those constructs as defined by this International Standard.

A conforming application intended to create topic maps must be able to export the information expressible using the constructs defined by this International Standard in conformance with the syntax of those constructs.

A conforming application intended to read and write topic maps must meet both of the above sets of requirements. If an application can import topic maps, has features to edit them, but cannot export altered topic maps in the interchange syntax defined by this International Standard, then it is not a conforming application.

NOTE 50 This International Standard constrains neither the uses to which topic maps can be put, nor the character of the processing that may be applied by a conforming application. This conformance clause is intended to guarantee that conforming topic maps can be understood to whatever degree conforming read-only applications are intended to understand them, and that the topic mapping information expressed using the topic map syntax will be preserved by conforming read/write applications (except to the extent that the users of read/write applications deliberately alter that information).

Annexes

A Annex A: Topic Maps Meta-DTD (Normative)

While the meta-DTD that appears here is normative, it is also possible for conforming applications to use a meta-DTD that is the same in all respects except for differences dictated by the need to use a different SGML declaration.

NOTE 1 For example, the meta-DTD will differ in certain syntactic details when used in XML (WebSGML) applications.

```

<!-- HyTime architectural support declarations -->

<?IS10744 ArcBase HyTime>
<!NOTATION HyTime
  PUBLIC "ISO/IEC 10744:1997//NOTATION AFDR ARCBASE
        Hypermedia/Time-based Structuring Language (HyTime)//EN"
>
<!ATTLIST #NOTATION HyTime
  ArcFormA NAME      HyTime
  ArcNamrA NAME      HyNames
  ArcSuprA NAME      sHyTime
  ArcIgnDA NAME      HyIgnD
  ArcDocF NAME       #FIXED HyDoc
  ArcDTD CDATA       "HyTime"
  ArcQuant CDATA     #FIXED "NAMELEN 12"
  ArcDataF NAME      #FIXED HyBridN
  ArcBridF NAME      #FIXED HyBrid
  ArcAuto (ArcAuto|nArcAuto) nArcAuto
  ArcOptSA NAMES     "base links locs"
  hyqcnt NUMBER      32
  base CDATA         "bos bosspec"
  locs CDATA         "agrovdef bibloc dataloc datatok grovplan listloc
                    mixedloc multloc nameloc nmsploc pathloc
                    pgrovdef proploc queryloc referatt refloc
                    reftype relloc spanloc treecom treeloc treetype"
  links CDATA        "varlink"
  exrefs NAME        exrefs
  manyanch NUMBER    #IMPLIED
>
<!NOTATION AFDRMeta
  PUBLIC "ISO/IEC 10744//NOTATION AFDR Meta-DTD Notation//EN"
>
<!ENTITY HyTime
  PUBLIC "ISO/IEC 10744//DTD AFDR Meta-DTD
        Hypermedia/Time-based Structuring Language (HyTime)//EN"
  CDATA AFDRMeta
>

<!--          HyTime common attributes          -->

<!-- The attribute form "HyTime common attributes"
      (common) declares the HyTime attributes shared by all
      topic map forms. -->

<!attlist
  -- Common --
  -- HyTime common attributes --
  -- HyTime Clause A.5.2, 7.8 --

```

```

#ALL
id          -- Unique identifier --
  ID
  #IMPLIED -- Default: None --
-- refloc -- -- Reference Location Address --
              -- HyTime Clause: 7.8 --

#ALL
loctype     -- Reference location addresses type --
              -- Each named attribute treated as if it were an
              -- IDREF to a location address element. --
              -- Constraint: The declared values of named
              -- attributes must be lexically compatible with
              -- their specified interpretation. --
              -- Note: The declared value CDATA always meets this
              -- requirement. --
  CDATA     -- Lexotype: (ATTORCON,("IDLOC"|"TREELOC"|
              "PATHLOC"|"RELLOC"|
              ("QUERYLOC",NOTATION)))+ --

  #IMPLIED -- Constant --
              -- Default: All references use SGML IDREFs, and each
              -- IDREF in an IDREFS attribute is considered
              -- separately --

  rflocsrc  -- Reference location source --
              -- Associates referential attributes with their
              -- location sources. --
  CDATA     -- Lexotype: (ATTORCON,ATTORCON)+ --
              -- Constraint: Attributes named must be referential
              -- attributes. --

  #IMPLIED -- Constant --
              -- Default: All referential attributes have this
              -- element as their location source. --

-- rflocspn -- -- Reference location span --
                -- HyTime Clause: 7.8 --

#ALL
rflocspn    -- Reference location span --
              -- Names pairs of referential attributes that
              -- address spans when both attributes are
              -- specified. --
  CDATA     -- Lexotype: (ATTORCON,ATTORCON)+ --
              -- Constraint: Attributes named must be referential
              -- attributes. --

  #IMPLIED -- Constant --
>

<!-- HyTime meta-DTD content model parameter entities -->

<!entity %
  HyCFC      -- HyTime context-free content --
              -- Note: %loc, %link, %resorce qualify but are used
              -- as meta-inclusions rather than meta-proper-
              -- subelements --
  "HyBrid"
>
<!entity %
  loc        -- Location address forms --
  "anchloc|bibloc|dataloc|fcsloc|linkloc|listloc|mixedloc|nameloc|
  nmsploc|pathloc|proploc|queryloc|relloc|treeloc"
>
<!entity %
  link       -- Hyperlink forms --
  "varlink"
>
<!entity %
  resbase    -- Base module resource forms --
  "bosspec"
>
<!entity %

```

```

    resloc          -- Location address module resource forms --
    "agrovdef|datatok|grovplan|pgrovdef"
>
<!entity %
    resorce          -- All resource architectural forms --
    "%resbase;|%resloc;"
>

<!--          HyTime document element form          -->

<!element
    HyDoc            -- HyTime document element --
                    -- HyTime Clause: 6.4 --
    - O
    (%HyCFC;)*
    +(%link;|%loc;|%resorce;)
-- OptionalAttributes [base]: bos, bosspcat --
-- OptionalAttributes [locs]: dgrvplan --
-- CommonAttributes [locs]: refloc, reftype, rflocspn --
>
<!attlist
    -- bos --          -- HyTime bounded object set --
                    -- HyTime Clause: 6.5.1 --
    (HyDoc)
    maxbos           -- Maximum bounded object set level --
                    -- Bounding level of HyTime bounded object set when
                    -- document is a hub or subhub. --
    NUMBER          -- Constraint: Depth of nested entities to include
                    -- in BOS (0=no limit, 1=hub only) --
    0
    boslevel        -- Bounded object set level --
                    -- Default BOS level used by data entities declared
                    -- in hub document. --
    NUMBER          -- Constraint: Depth of nested entities to include
                    -- in BOS (0=no limit, 1=this entity only) --
    #IMPLIED        -- Default: No HyTime BOS --
-- bosspcat --      -- BOS except specification attributes --
                    -- HyTime Clause: 6.5.3 --
    bosspec         -- Bounded object set exception specification --
                    -- Adjustments to be made to the bounded object
                    -- set. --
    IDREFS          -- Reference --
                    -- Reftype: bosspec+ --
                    -- Constraint: Must be internal reference --
    #IMPLIED        -- Default: No BOS exception specification --
-- dgrvplan --      -- HyTime document grove plan --
                    -- HyTime Clause: 7.1.4.1 --
    grovplan        -- Grove plan --
                    -- Grove plan for HyTime extended SGML document
                    -- grove --
    CDATA           -- Reference --
                    -- Reftype: grovplan --
    #IMPLIED        -- Default: HyTime default grove plan --
>

<!-- HyTime bounded object set control forms -->

<!-- HyTime BOS control data attributes -->

<!attlist #NOTATION
-- bosdatt --      -- HyTime BOS control data attributes --
                    -- HyTime Clause: 6.5.2 --
    #ALL
    boslevel        -- BOS level --
                    -- Bounded object set level for the entity --
    NUMBER          -- Constraint: Depth of nested entities to include
                    -- in BOS (0=no limit, 1=this entity only) --
    #IMPLIED        -- Default: Value of boslevel attribute of HyDoc
                    -- element. --

```

```

inbos          -- Include in BOS --
              -- Unconditional include in, or exclude from, BOS --
              (inbos|notinbos)
              #IMPLIED -- Default: Inclusion controlled by BOS level --
bosprerty     -- Bounded object set priority --
              -- Default BOS priority of objects in entity tree
              -- rooted at this entity. --
              (foregrnd|backgrnd)
              foregrnd
subhub        -- Is entity a subhub? --
              (subhub|nosubhub)
              nosubhub
>

<!-- HyTime bounded object set exception specification -->
<!element
  bosspec     -- Bounded object set exception specification --
              -- HyTime Clause: 6.5 --
              -- Used to affect the HyTime BOS by overriding the
              -- inclusion or exclusion and priority of the
              -- entities identified by the BOS path or paths
              -- given as content. --
  - 0
  (#PCDATA)   -- Lextype: ((ENTITY,(cname|literal)*|
              --                    (GRPO,ENTITY,(cname|literal)*,GRPC)+
              --
              -- Constraint: If parentheses are used, each
              -- parenthesized list is a separate BOS path. --
              -- Constraint: Each word or literal in a BOS path is
              -- the name of an entity declared in the entity
              -- identified by the previous word, literal, or
              -- entity name. --
  -- Attributes [base]: bosspec --
  -- Referrers [base]: HyDoc:bosspec --
>
<!attlist
  bosspec     -- Bounded object set exception specification --
              -- HyTime Clause: 6.5 --
  boslevel    -- BOS level --
              -- The BOS level from the last entity named in
              -- each specified BOS path to be affected by this
              -- bosspec. --
              NUMBER
              1
              -- Constraint: Depth of nested entities to include
              -- in BOS (0=no limit, 1=last entity only) --
  inbos       -- Include in BOS --
              -- Unconditionally include or exclude objects
              -- declared by the last entity named in each BOS
              -- path, to the BOS level specified by this
              -- bosspec's boslevel attribute. --
              (inbos|notinbos)
              #IMPLIED -- Default: BOS unaffected --
  bosprerty   -- Bounded object set priority --
              -- Unconditionally specify the BOS priority of
              -- objects declared by the last entity named in each
              -- BOS path, to the BOS level specified by this
              -- bosspec's boslevel attribute. --
              -- Note: The semantic of the bosprerty attribute is
              -- not affected by the value of the inbos attribute
              -- (that is, whether it is explicitly "inbos" or the
              -- value is implied). --
              (foregrnd|backgrnd)
              #IMPLIED -- Default: No priority change --
>

<!-- Topic Map specific architectural forms -->

```

```

<!entity %
  TMCFC          -- Topic map context-free content --
  "topic|assoc|facet|bosspec|addthms|TMBrid"
>
<!element
  TMBrid        -- Topic map bridge element --
  - O
  ANY
>

<!element
  topicmap      -- Topic map document element --
  -- Clause: 5.1 --
  - O
  (%TMCFC;)*
>
<!attlist
  topicmap
  HyTime        -- HyTime architectural form name --
  NAME
  HyDoc         -- HyTime document element. (This
  attribute definition is redundant; it
  appears here as an aid to
  understanding.) --
  addthms       -- Added themes --
  -- Themes to add to all scopes that govern
  the assignments of topic names,
  occurrences, and roles played in
  associations in this topic map
  document. --
  CDATA         -- Reference --
  -- Reftype: topic+ --
  #IMPLIED      -- Default: No themes added via this
  attribute. --
  -- bos --     -- HyTime bounded object set --
  -- HyTime Clause: 6.5.1 --
  maxbos        -- Maximum bounded object set level --
  -- Bounding level of HyTime bounded object
  set when document is a hub or
  subhub. --
  NUMBER        -- Constraint: Depth of nested entities to
  include in BOS (0=no limit, 1=hub only)
  --
  0
  boslevel      -- Bounded object set level --
  -- Default BOS level used by data entities
  declared in hub document. --
  NUMBER        -- Constraint: Depth of nested entities to
  include in BOS (0=no limit, 1=this
  entity only) --
  #IMPLIED      -- Default: No HyTime BOS --
  -- bosspcat -- -- BOS exception specification attributes
  --
  -- HyTime Clause: 6.5.3 --
  bosspec       -- Bounded object set exception
  specification --
  -- Adjustments to be made to the bounded
  object set. --
  IDREFS        -- Reference --
  -- Reftype: bosspec+ --
  -- Constraint: Must be internal reference
  --
  #IMPLIED      -- Default: No BOS exception specification
  --
  -- dgrvplan -- -- HyTime document grove plan --
  -- HyTime Clause: 7.1.4.1 --
  grovplan      -- Grove plan --
  -- Grove plan for HyTime extended SGML
  document grove --

```

```

        CDATA          -- Reference --
        #IMPLIED      -- Reftype: grovplan --
        #IMPLIED      -- Default: HyTime default grove plan --
<!element
  topic              -- Topic link --
                    -- Clause: 5.2.1 --
  - 0
  ( topname | occurs)*
>
<!attlist
  topic
  HyTime            -- HyTime architectural form name --
  (varlink|HyBrid)
  varlink          -- Constraint: varlink must be specified
                    when occurrences exist. If topic has no
                    occurrences, it must be declared as a
                    HyTime bridge element (HyBrid). --
  id               -- Unique identifier --
  ID
  #REQUIRED
  identity         -- Subject identity --
                    -- Reference to information (one or more
                    subject descriptors) that confers
                    understanding of the identity of the
                    subject of this topic link. --
  CDATA           -- Reference --
  #IMPLIED        -- Default: No subject descriptors; the
                    subject must be inferred from the
                    topic's characteristics. --
  types           -- Topic types --
                    -- Topics whose subjects are the classes
                    of topics of which this topic is an
                    instance. --
  CDATA           -- Reference --
  #IMPLIED        -- Reftype: topic+ --
                    -- Default: No class-instance topic
                    associations are established via this
                    attribute. --
                    -- Note: Some might still be specified by
                    topic association links, however. --
  scope           -- Scope --
                    -- The themes that are added to the scopes
                    of all the names and occurrences
                    specified by this topic link. --
  CDATA           -- Reference --
  #IMPLIED        -- Reftype: topic+ --
                    -- Default: No themes are added by this
                    attribute. --
  linktype        -- Hyperlink type --
  NAME
  #IMPLIED        -- Default: Generic identifier --
<!element
  topname           -- Topic name --
                    -- Clause 5.2.2--
  0 0
  (basename+, dispname*, sortname* )
                    -- If dispnames or sortnames are not
                    specified, applications use basenames
                    for display and sorting purposes. --
>
<!attlist
  topname
  scope           -- Scope --
                    -- Reference to a set of themes (topic
                    links) to be added to the scopes of the
                    name characteristics specified by the
                    contained basename, dispname, and

```

```

        sortname elements.  Scopes are sets of
        themes that collectively define the
        limited context within which
        characteristics are validly applicable
        to the topic. --
        CDATA          -- Reference --
        #IMPLIED      -- Reftype: topic+ --
        #IMPLIED      -- Default: No themes are added via this
        attribute. --
    >
<!element
  (basename | sortname)
    -- Base name --
    -- and --
    -- Name to be used as sort key --
    - 0
  (#PCDATA)          -- String to be used as name --
>
<!element
  dispname          -- Display name --
  - 0
  (#PCDATA|TMbrid)*
    -- String (or notation data) to be
    displayed as name --
>
<!attlist ( basename | sortname | dispname)
  scope            -- Scope --
                  -- References to a set of themes (topic
                  links) to be added to the scope of the
                  name characteristic specified in the
                  content. --
  CDATA           -- Reference --
  #IMPLIED       -- Reftype: topic+ --
  #IMPLIED       -- Default: No themes are added via this
  attribute. --
>
<!element
  occurs           -- Topic occurrence --
                  -- Clause: 5.2.3 --
  - 0
  (%loc;)*
>
<!attlist
  occurs
  HyTime          -- HyTime architectural form name --
  NAME
  #FIXED
  anchspec
  scope          -- Scope --
                -- Reference to themes that are added to
                the scope within which the occurrences
                are applicable to the topic
                characterized by the containing topic
                link.--
  CDATA         -- Reference --
  #IMPLIED     -- Reftype: topic+ --
  #IMPLIED     -- Default: No themes are added to the scope
  by means of this attribute. --
  occr1        -- Occurrence role name --
                -- Note: Not displayed for the topic map
                user if the topic referenced by the
                type attribute has displayable
                characteristics within the user's
                scope. --
  NAME
  #IMPLIED     -- Default: GI of element is treated as
  occurrence role name. --
  type         -- Occurrence role type --

```

```

-- Reference to the topic that names
-- and/or otherwise characterizes the
-- occurrence role. The characteristics
-- of the referenced topic, if
-- appropriate, will be displayed to the
-- user instead of the value of the occrl
-- attribute. --
CDATA -- Reference --
#IMPLIED -- Reftype: topic --
-- Default: No topic characterizes the
-- occurrence role, unless this element is
-- an occurrence (with an occurrence role
-- whose meaning is instance) of a topic
-- whose subject is the nature of the
-- occurrence role. The value of the
-- occrl attribute will be displayed as
-- the occurrence role name. --
linktrav -- Hyperlink traversal rules --
-- Traversal between anchors of hyperlinks:
-- A any traversal or departure (EID)
-- D departure after internal arrival
-- E traversal after external arrival
-- I traversal after internal arrival
-- N no traversal after internal arrival
-- P no internal arrival
-- R return traversal after internal arrival --
NAMES -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|
-- "I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
A
listtrav -- List traversal rules --
-- Traversal between members of list anchors:
-- A adjacent (both left and right) traversal
-- L left traversal
-- N no traversal
-- R right traversal
-- W wrapping traversal --
NAMES -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
N -- Default: Show the whole list --
multmem
(single|list|corlist)
list
emptyanc
(error|noterror)
error
HyNames
CDATA
"anchrole occrl"
>
<!element assoc -- Association link --
-- Clause: 5.3.1 --
- 0 (assocrl)+ >
<!attlist assoc
HyTime -- HyTime architectural form name --
NAME
#FIXED
varlink
scope -- Scope --
-- Reference to themes that are added to
-- the scope within which the association
-- is applicable. --
CDATA -- Reference --
-- Reftype: topic+ --
#IMPLIED -- Default: Scope is unconstrained. --
linktype -- Hyperlink type. --
-- Mnemonic name for the association
-- type. --
-- Note: Not displayed for the topic map
-- user if the topic referenced by the

```



```

type attribute has displayable
characteristics within the user's
scope. --
    NAME
    #IMPLIED -- Default: Generic identifier --
type -- Association type --
-- Topic whose subject is the class of
association of which this association
is an instance. --
    CDATA -- Reference --
-- Reftype: topic --
    #IMPLIED -- Default: No type is specified by this
attribute. --
-- Note: A type might exist by virtue of
the fact that this association link is
an occurrence (where the occurrence
role means "instance") of a topic whose
subject is the nature of the
association, however. --
>
<!element
  assocrl -- Association role --
-- Clause: 5.3.2 --
  - 0
  (%loc;)+ -- Reftype: topic+ --
>
<!attlist
  assocrl
    HyTime -- HyTime architectural form name --
    NAME
    #FIXED
    anchspec
  anchrole -- Anchor role --
-- Note: Not displayed for the topic map
user if the topic referenced by the
type attribute has displayable
characteristics within the user's
scope. --
    NAME
    #IMPLIED -- Default: GI of element is treated as
anchor role. --
  type -- Association role type --
-- Reference to the topic that names
and/or otherwise characterizes the
association role. The characteristics
of the referenced topic, if
appropriate, will be displayed to the
user instead of the value of the
anchrole attribute. --
    CDATA -- Reference --
-- Reftype: topic --
    #IMPLIED -- Default: No topic characterizes the
association role, unless this element
is an occurrence (with an occurrence
role whose meaning is instance) of a
topic whose subject is the nature of
the association role. The value of the
anchrole attribute will be displayed as
the association role name. --
  linktrav -- Hyperlink traversal rules --
-- Traversal between anchors of hyperlinks:
A any traversal or departure (EID)
D departure after internal arrival
E traversal after external arrival
I traversal after internal arrival
N no traversal after internal arrival
P no internal arrival
R return traversal after internal arrival --

```

```

    NAMES          -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|"
                    "I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
    A
listtrav          -- List traversal rules --
                  -- Traversal between members of list anchors:
                  A adjacent (both left and right) traversal
                  L left traversal
                  N no traversal
                  R right traversal
                  W wrapping traversal --
    NAMES          -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
    N              -- Default: Show the whole list --
multmem
  (single|list|corlist)
  list
emptyanc
  (error|noterror)
  error
>
<!element
  addthms          -- Themes to be added --
                  -- (To scopes specified by topic map
                  documents and/or by topic links and/or
                  association links.) --
                  -- Clause: 5.4 --
  - 0
  (TMBrid)*       -- No content defined by the Topic Maps
                  architecture --
>
<!attlist
  addthms          -- Themes to be added --
                  -- Clause: 5.4 --
  addthems        -- Added themes --
                  -- Themes to be added to the scopes
                  specified by the tmdocs and cassign
                  attributes --
  CDATA           -- Reference --
                  -- Reftype: topic+ --
  #REQUIRED
  tmdocs          -- Topic map document entities --
  ENTITIES        -- Constraint: Must be one or
                  more document entities of
                  topic map documents. --
  #IMPLIED
  cassign         -- Characteristic assigners --
                  -- Elements that assign characteristics to
                  topics. The themes specified by the
                  addthms attribute are to be added to
                  the scopes within which the
                  characteristics they specify are
                  regarded as valid --
  CDATA           -- Reference --
                  -- Reftype: (topic | topname | basename |
                  dispname | sortname | occurs | assoc)+
                  --
  #IMPLIED
>
<!element
  facet           -- Facet link --
                  -- Clause: 5.5.1 --
  - 0
  (fvalue)+
>
<!attlist facet
  HyTime          -- HyTime architectural form name --
  NAME

```

```

    #FIXED
    varlink
linktype      -- Hyperlink type. --
              -- Mnemonic name for the property (facet
              type). --
              -- Note: Not displayed for the topic map
              user if the topic referenced by the
              type attribute has displayable
              characteristics within the user's
              scope. --

    NAME
    #IMPLIED   -- Default: Generic identifier --
type         -- Facet type --
              -- Topic whose subject is the property of
              the property/value pair(s) being
              assigned to the anchor(s). --

    CDATA      -- Reference --
              -- Reftype: topic --

    #IMPLIED   -- Default: No facet type topic is
              specified by this attribute. --
              -- Note: A facet type topic might exist by
              virtue of the fact that this facet link
              is an occurrence (where the occurrence
              role means "instance") of a topic whose
              subject is the nature of the property,
              however. --

>
<!element
  fvalue      -- Facet value --
              -- Clause: 5.5.2 --

  - 0
  (%loc;)*
>
<!attlist
  fvalue
  HyTime     -- HyTime architectural form name --
    NAME
    #FIXED
    anchspec
  facetval   -- Facet value name --
              -- Token is value of property being
              assigned. --

    NAME
    #IMPLIED   -- Default: Facet value name is GI of
              element. --
type        -- Facet value type --
              -- Reference to a topic whose subject is
              the significance of the facet value
              name. --

    CDATA      -- Reference --
              -- Reftype: topic --

    #IMPLIED   -- Default: No facet value type topic is
              specified by this attribute. --
              -- Note: A facet value type topic might
              exist by virtue of the fact that this
              fvalue element is an occurrence (where
              the occurrence role means "instance")
              of a topic whose subject is the
              significance of the facet value name,
              however. --

  linktrav   -- Hyperlink traversal rules --
              -- Traversal between anchors of hyperlinks:
              A any traversal or departure (EID)
              D departure after internal arrival
              E traversal after external arrival
              I traversal after internal arrival
              N no traversal after internal arrival
              P no internal arrival

```

```

                                R return traversal after internal arrival --
NAMES                          -- Lextype: ("A"|"EI"|"ER"|"ED"|"EN"|"EP"|"ERD"|"
                                "I"|"ID"|"D"|"N"|"P"|"R"|"RD") --
A
listtrav                        -- List traversal rules --
                                -- Traversal between members of list anchors:
                                A adjacent (both left and right) traversal
                                L left traversal
                                N no traversal
                                R right traversal
                                W wrapping traversal --
NAMES                          -- Lextype: ("A"|"AW"|"L"|"LW"|"N"|"R"|"RW") --
N                               -- Default: Show the whole list --
multmem
  (single|list|corlist)
  list
emptyanc
  (error|noterror)
  noterror
HyNames
  CDATA
  "anchrole facetval"
>
```

B Annex B: Example Architectural Support Declaration for the Topic Maps Architecture (Informative)

The following is an example of a topic map architectural support declaration as it might appear in a topic map document that is expressed using SGML.

NOTE 2 The following example conforms to Annex A.3 of ISO/IEC 10744:1997.

```
<!NOTATION TOPICMAP
  PUBLIC "ISO/IEC 13250:1999//NOTATION AFDR ARCBASE
        Topic Maps//EN"
>
<!ATTLIST #NOTATION TOPICMAP
  ArcFormA NAME      TopicMap
  ArcNamrA NAME      TMNames
  ArcSuprA NAME      sTopMap
  ArcIgnDA NAME      TMIgnD
  ArcDocF  NAME      #FIXED topicmap
  ArcDTD   CDATA     "TMDTD"
  ArcQuant CDATA     #FIXED "NAMELEN 12"
  ArcDataF NAME      #FIXED TMBridN
  ArcBridF NAME      #FIXED TMBrid
  ArcAuto  (ArcAuto|nArcAuto) nArcAuto
>
<!NOTATION AFDRMeta
  PUBLIC "ISO/IEC 10744//NOTATION AFDR Meta-DTD Notation//EN"
>
<!ENTITY TMDTD
  PUBLIC "ISO/IEC 13250:1999//DTD AFDR Meta-DTD
        Topic Maps//EN"
  CDATA AFDRMeta
>
```

The following is an example of topic map architectural support declarations as they might appear in a topic map document expressed using WebSGML or XML.

NOTE 3 The following example conforms to Annex A.3 of ISO/IEC 10744:1997, as amended via Amendment 1 in 1998.

```
<?IS10744:arch
  name="TopicMap"
  public-id="ISO/IEC 13250:1999//NOTATION AFDR ARCBASE
            Topic Maps//EN"
  dtd-public-id="ISO/IEC 13250:1999//DTD AFDR Meta-DTD
               Topic Maps//EN"
  dtd-system-id="/etc/architectures/TM.mtd"
  form-att="TopicMap"
  renamer-att="TMNames"
  suppressor-att="sTopMap"
  ignore-data-att="TMIgnD"
  doc-elem-form="topicmap"
  bridge-form="TMBrid"
  auto="nArcAuto"
?>
```