TOPIC MAP EXPLORER

by

AYŞENUR ÖZBOYACI

November 2005

TOPIC MAP EXPLORER

by

AYŞENUR ÖZBOYACI

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

November 2005

Istanbul, Turkey

APPROVAL

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Kemal FİDANBOYLU
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist Prof. Dr. Atakan KURT
Supervisor

Examining Committee Members

Assist Prof. Dr. Atakan KURT

Assist Prof. Dr. Veli HAKKOYMAZ

Assoc Prof. Dr. Erkan İMAL

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

Assoc. Prof. Dr. Nurullah ASLAN
Director

November, 2005

TOPIC MAP EXPLORER

AYŞENUR ÖZBOYACI

M.S. Thesis – Computer Engineering
September 2005

Supervisor: Assist. Prof. Atakan KURT

ABSTRACT

In this thesis, we have developed a text based topic map exploration technique based on type hierarchies. A type hierarchy is a hierarchical organization of a set of types with super/sub type relationships among them. This hierarchy provides a simple intuitive way to explore topic maps. The type hierarchy allows one to navigate easily in a topic map at different levels. A topic map contains the following entities: topics, associations, roles and occurrences. For each one of these entity types, we define four separate type hierarchies in a topic map, namely topic-type hierarchy, association-type hierarchy, role-type hierarchy, occurrence-type hierarchy. The inner nodes in these hierarchies are entities types, leaf nodes are entities or objects. We also define type-level (intermediate) entity hierarchies and object (local) level entity hierarchies for inner and leaf nodes in the 4 main (global or topic map) hierarchies to represent the set of entities related to a specific node in the form of a hierarchy itself. The concepts here are demonstrated with a prototype hierarchical topic map explorer written in Java.

Keywords: Topic Maps, Topic Map Tool, Type Hierarchies

KONU HARİTALARI METİNSEL GÖSTERİMİ

AYŞENUR ÖZBOYACI

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Eylül 2005

Tez Yöneticisi: Yrd. Doç. Dr. Atakan KURT

ÖZ

Bu çalışmada, *tür hiyerarşileri* üzerine bina edilmiş bir *konu haritası* gezinti tekniği geliştirilmiştir.Tür hiyerarşisi aralarında üst-alt tür ilişkisi bulunan bir tür seti içerisindeki tür ve nesneleri bir ağaç hiyerarşisi halinde organize etmek demektir. Böyle bir hiyerarşi konu haritalarını gezinti ve anlama için basit ve tabi bir yol sağlar. Böyle bir hiyerarşi ile konu haritasını değişik düzeylerde gezmek mümkün olabilir. Bir konu haritasında şu nesneler vardır: konular, ilişkiler, roller ve geçtiği-yerler. Bu nesneler için 4 adet nesne tipi konu haritalari içinde tanımlanabilir: konu türleri, ilişki türleri, rol türleri ve geçtiği-yer türleri. Bu türleri kullanarak ilgili hiyerarşiler tanımlanır. Bu hierarşilerin iç düğümlerinde türler, yaprak düğümlerinde nesneler bulunur.Bu konu haritası bazındaki (*genel düzey*) hiyerarşilere ek olarak, tür bazında (*ara düzey*) ve nesne bazında (*alt yada lokal düzey*) hiyerarşiler de tanımlanmıştır. Bu hiyerarşilerde seçilen nesne ile ilgili tüm diğer bilgiler ayrı ayrı ara düzey yada lokal düzey hiyerarşileri olarak verilir. Bu konseptler bir konu haritası gezgini gerçekleştirilerek tekniğin pratiğe uygulanabilirliği gösterilmiştir.

Anahtar Kelimeler: Konu Haritaları, Konu Haritaları Gezgini, Tür Hiyerarşileri

To computer science

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor Assist. Prof. Atakan KURT for his guidance and insight throughout the research.

Thanks go to the jury members for their valuable suggestions and comments.

I express my thanks and appreciation to my parent and husband for their support, understanding, and patience. I am also thankful to my friends who made my stay at university a memorable and valuable experience.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Topic maps aim to represent the information in a specific area to users in a structured way. Anything in the world can be represented as a topic in a topic map which consists of topics associated among them. The information about topics (occurrences) may come from a web page or a database. This relationship between the source and the topic is represented by the URI links in topic maps. The information in the topic maps is given in a standard called XTM (XML Topic Maps). Topic map creates a structured collection of information that user can navigate, query and understand. Topic map document has a collection of information and may be large in size. Because of this, for large documents navigation in a topic may be able to be difficult. Topic map is one concept for representing information. Topic maps are referred at (Biezunski, M. et al 2001, Freese E. 2002, Freese E 2003, Garshol L.M. 2002, Garshol L.M. 2004, Kal Ahmed 2000, Kal Ahmed 2001, Ksiezyk R. 1999, Ksiezyk R. 2000, Pepper S. 2002, http://www.topicmaps.org/xtm/1.0/). Other concepts are semantic web and ontology as referred at (Falkovych K 2003, Granmo G. O. 2000, Le Grand et al 2002, Lin X. et al 1991) and (Kalyanpur A. et al 2004). Artificial intelligent and similar fields are trying to create meta-data layer on the huge amount of information in the web.

A topic map document consists of four main entities: *topics*, *associations*, *roles* and *occurrences*. A *topic* can be any thing; conceptual or real. In a topic map document, all entities (associations, roles, occurrences) is considered topic where appropriate or necessary. Topic may represent resource that can be retrieved from the computer or it can be anything in the world. “ali” and “ahmet” can be topics in the topic map document. Topic is the central concept of topic maps. It represents the concepts or objects in the subject of the topic map.

Information resources linked to topics are defined as *occurrences* of topics. Topics may be related zero or more resources outside the topic map. Let’s say that there is a web page related to “ahmet” then the web page is a resource of the topic “ahmet”. In the topic map document web page is usually defined with a URL.

Relationships between those topics are defined using a*ssociations*. Associations may be defined between any numbers of topics. Each topic in an association is called a member of that association. There may be an association between topics “ali” and “ahmet”, let’s call that association “education”. In associations topics have roles or play roles. In our “education” association there are two topics so they have roles. Topic “ali” has the role “student” and topic “ahmet“ has the role “teacher”. “ali” and “ahmet” are the members of the “education” association. A topic map document mainly formed with these four kinds entities, there are other concepts related to topic map which are not discussed here. The entities we consider in this study are: Topics, associations, occurrences may have parent entity/entities.

```
<topicmap xmlns:xlink="http://www.w3.org/1999/xlink"
commonThemes="mary">
   <topic id = "person" />
   <topic id = "course" />
   topic id = "academic" />
   <topic id = "advice" instanceOf = "teach" />
   <topic id = "advisor" instanceOf = "academic" />
   <topic id = "mary" instanceOf = "instructor" />
   <topic id = "terry" instanceOf = "instructor" >
     <occurs type = "logo"
href="http://www.anywhere.com/~terry/me.gif" />
     <occurs type = "magazine" href =
"http://www.magazines.com/~terry" />
   </topic>
   <topic id = "jane" instanceOf = "instructor">
     <occurs type = "paper"
href="http://www.somewhere.com/~jane/papaers.html" />
     <occurs type = "webpage" href = "http://www.somewhere.com/~jane"
/>
   </topic>
   <assoc id ="zero" instanceOf = "advice" >
     <assocrl xlink:href="#mary" role="undergradadvisor"/>
     <assocrl xlink:href="#john" role="undergradst"/>
   </assoc>
   <assoc id = "education" instanceOf = "advice" >
     <assocrl xlink:href="#jane" role="advisor"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
</topicmap>
```

Figure 1.1 A Sample TM Document

A sample topic map document looks like in the Figure 1.1. Topic map has a tag called ***<topicmap>***. This tag encloses the entire topic map. Topics have ***id***s and ***instanceOf*** parameter. *id* represents the name of the topic, *instanceOf* parameter represents the parent topic/type of the topic. This relationship among types illustrates the idea behind hierarchies used in this study. Some topics have occurrences that means where the topic occurs (database, web, etc). Occurrences have the ***type*** parameter and ***href*** parameter. *type* parameter represents the type of the occurrences and the *href* parameter represents the name or the URI of the occurrences. Associations are relations between the topics. Associations have ***id*** and ***instanceOf*** parameter. Those parameters have the same role as in the topic tag. Each association has ***assocrl*** element. This element has ***xlink*** and ***role*** parameters. *xlink* parameter gives the name of the member topic and the *role* parameter represents the role of the topic in the association.

In Figure 1.1, “course” is a topic and it has no <instanceOf> parameter so this topic has no type or a parent node in other words. Topic “advisor” has <instanceOf> parameter. This indicates that this topic has a type called “academic” which is also a topic. In the document (topic map), “course” and the “advisor” are topics. “course” has no type but “advisor” has type called “academic” that it is also a topic in the document. Between these topics there is an association called “education”. This relation also has an <instanceOf> parameter. This relation has a type also called “advice” and this type is also a topic and it also has a type called “teach”. We can see that in the document everything is a topic and they are related/connected to each other. “education” association has two members called “jane” and “cs101” and they have roles in the association. “jane”s role is “advisor” and “cs101”s role is “course”. We see that in the “education” relation “jane” is “advisor” and “cs101” is “course”. In the document “jane” and “cs101” are also topics. According to the document in the Figure 1.1, we see that “instructor” topic is type of two different topics so topic “instructor” has two children called “mary” and “terry”. Topic “terry” has occurrence in type “logo” that is also a type in the document. Topic “terry” has resources in type “logo”. The location of the occurrence is given in the *href* parameter so it is a web page.

A topic map document may consist of thousands of topics. Because of this reason the exploration of the topic map has become an important subject. There are some exploration techniques. All they want to make the exploration of the topic map is simplicity. This will help the user to navigate the topic map document easily.

As an example, *Topic Map Visualization* technique is developed by Kerziban Mumcu who was a graduate student of Fatih University. That technique allows users to visualize and navigate a topic map. In the study the topic map document is graphically represented using *higraphs* as shown in Figure 1.2. Set of elements are represented with some special relations on them.

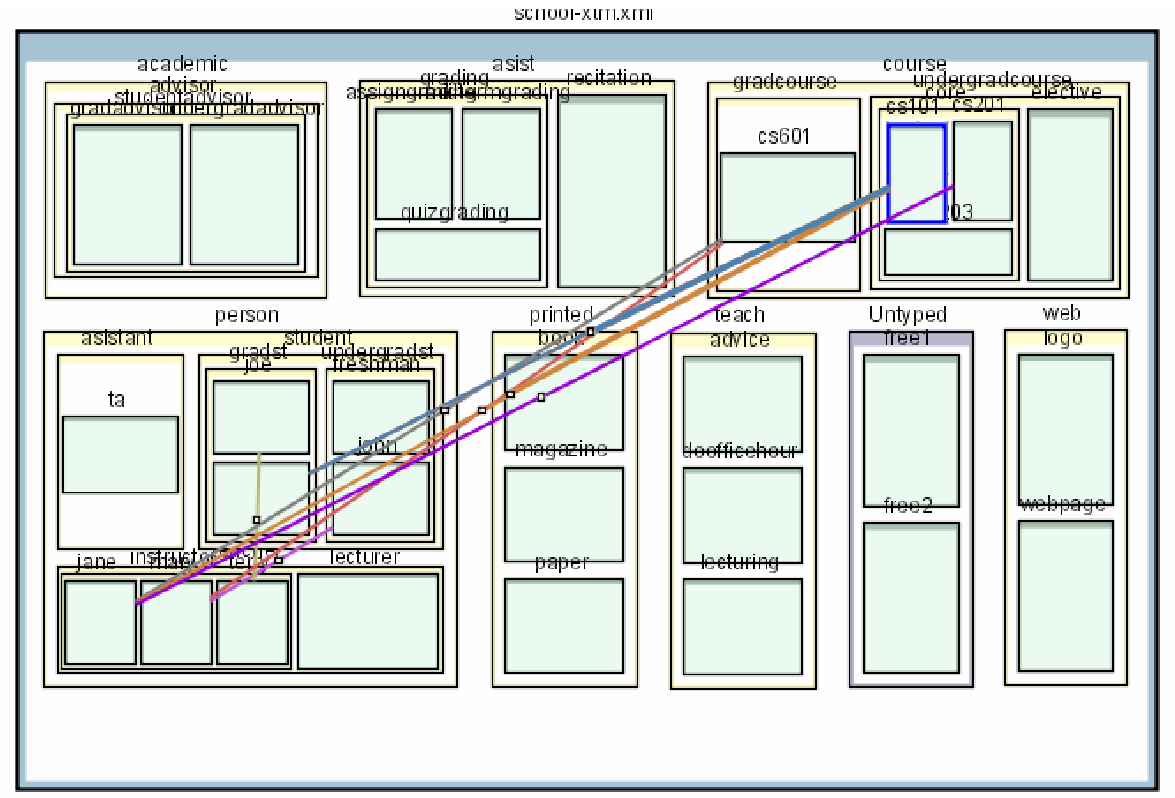


Figure 1.2 Topic Map Visualization

There exists some more related works such as (Garshol L. M. 2001, http://www.xm.co.nz/op/tm.htm, Kunz C. et al 2002, Le Grand B. et al June-2000, Le Grand B. et al December-2000, Spoerri A. 1993, http://www.techquila.com/practical_intro.html, http://www.techquila.com/tmsinia.html, Wrightson A. 2001, Mumcu K. 2005, http://protege.stanford.edu/)

In our study, we define the topic map model first. As we mentioned before, topics have the instanceOf parameter that shows the parent topic of the existing topic. This shows that there is a hierarchy (parent/child) between the topic elements. Each topic that is parent of another topic is called ***type object/entity/node*** and each topic that has no child called ***leaf object/entity/***node. This structure is same for the other entities that we mentioned (association, occurrences and roles). If entities have an instanceOf parameter, this shows that the topics in the instanceOf parameter are the (super) *types* of the current entities. If those entities have no child nodes then these current entities are called *leaf object.* If we consider the document as graph, then the leaf nodes are the objects and the inner nodes are the types in an analogy to the object oriented class hierarchy in which there is a hierarchy among a set of classes (inner nodes or types) and objects (leaf nodes or objects). Associations may have topic types or association types. An association can be a type of the other association. Occurrences are also same like this.

In our exploration we divide the topic map into four type hierarchy (**TTH**, topic type hierarchy; **ATH**, association type hierarchy; **OTH**, occurrences type hierarchy and **RTH**, role type hierarchy). Each hierarchy shows the parent/child relationships among respective entities in the topic map document. This parent/child relationship is also called in the study super/sub types. Super type(s) are the parent of the selected element (parent in the related upper levels) and the sub type(s) are the child elements of the selected node (parent in the related lower levels). This makes the user to navigate in the document easily. User can see all the hierarchies of the selected node (topic, association, occurrences or role). Shortly hierarchy in the document shows the super/sub type relationship. In the topic map the root element is the <topicmap>. This element is the root of the each hierarchy. This hierarchy is a tree structure rooted at <topicmap> tag.

The rest of the thesis is organized as follows: In Chapter 2 we discuss topic map and XTM syntax. We present the data model in Chapter 3. We introduce the hierarchical topic map explorer tool in Chapter 4. Chapter 5 is reserved for conclusions.

CHAPTER 2
TOPIC MAP AND XTM

2.1 TOPIC MAPS

The internet has grown so much that finding specific information becomes more difficult. This makes the querying or the search of the information more difficult. The search engines are the main resources to search. In the books information is indexed to be reached easily. This indexing idea produced the idea of the topic maps. In the computer society, topic maps, semantic web and ontologies are becoming an important technology. Topic map is like a meta data structure. It represents the main concepts anre relationships among them. This makes the information indexed. And this makes the query and navigation more sensible.

Topic map is a standard (ISO) for information representation and usage. This standard is ISO (the International Organization for Standardization) / IEC (the International Electrotechnical Commission) 13250:2003. Topic maps are created and organized in this standard.

A topic map can be summarized as in the Figure 2.1. As seen in the figure, topic map document has topics that can be anything, associations that are the relations of the topics (also associations may have relationship) and associations are shown by the arrows in the Figure 2.1. Associations are fundamentally different from the rest of the three entities in the topic map. Because they are defined between two or more topics and each of those topics have a role in associations. Occurrences give the URI of the topics resource. Those can be a type or object according to the place they are in. If we think the topic map as a tree, the leaf nodes are the objects that have no child and the inner nodes are the type nodes that have child/children. The type nodes are not the type of only one topic. A type node may have more then one child and this makes the topic map document much bigger and hierarchical. This hierarchy helps us to navigate and query the document.

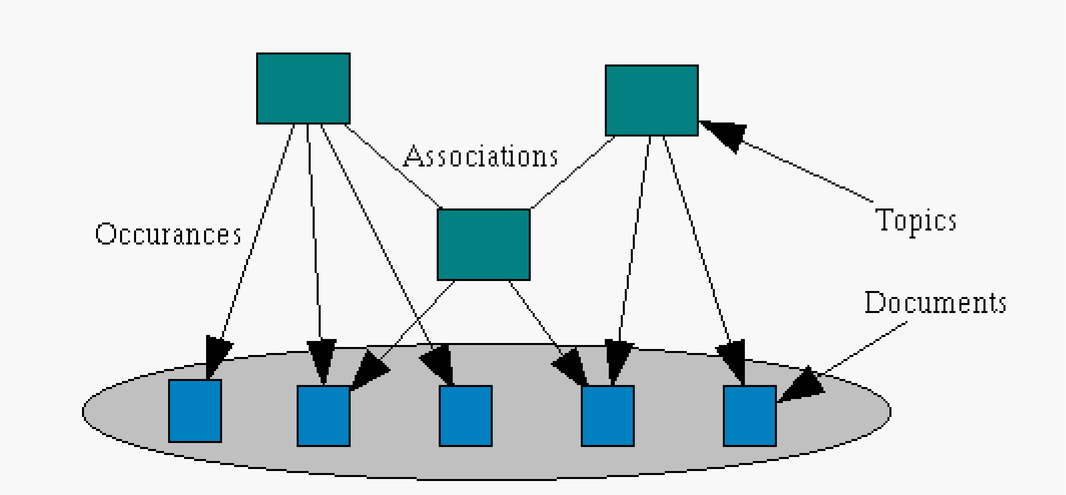


Figure 2.1 A Sample TM Document (http://www.xm.co.nz/op/tm.htm)

In 2001, TopicMaps.org published its XML Topic Maps (XTM) 1.0 specification. XTM is the main topic map syntax. Nearly every topic map tool uses this syntax.

<topic>

Topics can be everything. A topic may refer to a person, a web page, a database, a concept or something else. Those topics give you the idea what the topic map is about. Mainly it can be said the topics are the heart of the topic map document. Topics are categorized according to their type. This type shows the topic's parent node. This gives the hierarchy of the topic entity. In the topic map document topics are defined in the tag *<topic>*.

There are some properties for topic entity that we use in our study. These are the id, *instanceOf* and *occurrences* of the topics.

<id>

id represents the name of the topic.

<occurrence>

Occurrences show the relationship between topic and the resources. Resources may be a value inside or an external resource. Occurrences may be thought as associations. In this association one participant is the topic and the other one is the resource.

Occurrences may have parent nodes or types that are topics. According to this parent/child relationship we get the hierarchy of the occurrences entity.

<instanceOf>

InstanceOf is topic parameter in the topic map. This parameter shows the parent node of the topic. This parameter may take more than one value. Mainly, this parameter shows us the parent node list of the topic.

<association>

Associations are the relationships between the topics. Associations may have instanceOf or type/parent node. This shows that the associations may have the hierarchy by itself. This establishes the idea of the relationship of the topics. Topics that are related in the associations are the *member* of the associations. Each member has a *role* in the association.

<member> and <role>

In the associations, there are member and the role parameter. Member defines the participant topic of the association. The role parameter defines what the members do in the association.

2.2 XTM

As we said, topic map is standard (ISO) for information representation and usage. This standard is ISO (the International Organization for Standardization) / IEC (the International Electrotechnical Commission) 13250:2003.

In 2001, TopicMaps.org published its XML Topic Maps (XTM) 1.0 specification. XTM is the main topic map syntax. XTM DTD syntax is the mostly used syntax. Nearly all topic map tools use this syntax.

An XTM document is an XML document that defines a topic map and the properties of it. An XTM DTD lists the elements that are allowed in the XTM document, their properties and how they are connected. There are 19 element types are listed in the XTM document, but in the study we do not need all of them. The lists of the elements are topicMap, topic, instanceOf, subjectIdentity, topicRef, subjectIndicatorRef, baseName, baseNameString, variant, variantName, parameters, occurrence, resourceRef, resourceData, association, member, roleSpec, scope and mergeMap. We define only the elements/tags we used in this study.

CHAPTER 3
DATA MODEL

3.1 TYPE HIERARCHIES

In the topic map document there are four types of entities as we have mentioned before: topic, association, role and occurrence. Navigation or querying in the topic map is difficult because the topic map document is generally big and complicated. Topic maps can be modeled as a graph. Since we deal with hierarchies we represent topic maps with trees.. The root of the document is the <topicmap> tag. In the tree, the inner nodes are called *immediate* and the leaf nodes are called *local*. In the topic map document entities have the parent/child or super/sub type relationship among them. According to this, the graph represents a hierarchy among four entities. In our study, we divide the hierarchies into four separate hierarchies. This will give us a general idea of the document and the relationships. While querying, navigating or moving on the document over those four hierarchies will be much easier and quicker. For example, a query or navigation inside associations over this hierarchy will be quick because in the hierarchy there are only associations. There are types and object for four entities:

- *topic type (TT) and topic object (TO)* for topics,
- *association type (AT)* for associations,
- *role type (RT)* for roles
- *occurrences type (OT)* for occurrences.

We create four hierarchies called *topic map level* or *global* hierarchy

- *topic type hierarchy (TTH),*
- *association type hierarchy (ATH),*
- *role type hierarchy (RTH)*

- *occurrences type hierarchy (OTH).*

We define hierarchies in type (intermediate) level and object (local) level as well to provide a compact and easy navigation within the context of a chosen type or object. The overall navigation paths are shown in Table 3.1.

	Topics	Associations	Occurrences	Roles
Global (Topic Map) Level	TTH	ATH	RTH	OTH
Type (Intermediate) Level	TTH	ATH	RTH	OTH
Object (Local) Level	TTH	ATH	RTH	OTH

Table 3.1 Overall Navigation Paths

As shown in Table 3.1, there is 4 by 3 different hierarchies created at 3 different levels for 4 different kinds of entities.

While global hierarchy represents all entities in the topic may, type and object level hierarchies represent specialized hierarchies for a specific type or object in the topic map. The idea is to combine global, intermediate and local hierarchies in a single tree for each kind of entity for intuitive and easy navigation.

3.2 TOPIC MAP LEVEL (GLOBAL) HIERARCHIES

In this section we illustrate global (topic map level) topic, associations, occurences and role hierarchies in more details using the sample xtm document in Figure 3.1.Since we have explained how the hierarchies are built conceptually above we simply illustrate the idea using the sample document. The global level hiearchies as shown in figures Figure 3.2, Figure 3.3, Figure 3.4, Figure 3.5.

```
<topicmap xmlns:xlink="http://www.w3.org/1999/xlink" commonThemes="mary">
   <topic id = "course" />
   <topic id = "teacher" />
   <topic id = "instructor" instanceOf = "teacher" />
   <topic id = "printed" />
   <topic id = "web" />
   <topic id = "paper" instanceOf = "printed" />
   <topic id = "magazine" instanceOf = "printed" />
   <topic id = "logo" instanceOf = "web" />
   <topic id = "webpage" instanceOf = "web" />
   <topic id = "terry" instanceOf = "instructor" >
     <occurs type = "logo" href="http://www.anywhere.com/~terry/me.gif" />
     <occurs type = "magazine" href = "http://www.magazines.com/~terry" />
   </topic>
   <topic id = "jane" instanceOf = "instructor">
     <occurs type = "paper" href="http://www.where.com/~jane/papaers.html" />
     <occurs type = "webpage" href = "http://www.somewhere.com/~jane" />
   </topic>
   <topic id = "asistant" /> <topic id = "ta" instanceOf = "asistant" />
   <topic id = "asist" />   <topic id = "recitation" instanceOf = "asist" />
   <topic id = "grading" instanceOf = "asist" />
   <topic id = "assigngrading" instanceOf = "grading" />
   <topic id = "teach" />   <topic id = "lecturing" instanceOf = "teach" />
   <topic id = "advice" instanceOf = "teach" />
   <topic id = "lecturer" instanceOf = "teacher" />
   <topic id = "academic" /> <topic id = "advisor" instanceOf = "academic" />
   <assoc id = "iki" instanceOf = "advice" >
     <assocrl xlink:href="#jane" role="advisor"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "uc" instanceOf = "assigngrading" >
     <assocrl xlink:href="#sue" role="ta"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "bes" instanceOf = "recitation" >
     <assocrl xlink:href="#sue" role="ta"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "alti" instanceOf = "teach" >
     <assocrl xlink:href="#jane" role="teacher"/>
     <assocrl xlink:href="#cs201" role="course"/>
   </assoc>
      <assoc id = "yedi" instanceOf = "lecturing" >
     <assocrl xlink:href="#mary" role="lecturer"/>
     <assocrl xlink:href="#cs601" role="course"/>
   </assoc>
</topicmap>
```

Figure 3.1 XTM Document

The root element in these hiearchies covers all types that do not have a super type. Thus the root element is an over encompassing super type for all types in the topic map. The lowest level type i.e. the types that do not have any subtypes or the types that only have objects are called atomic types in the figures.

```
-TTH
 -courses
 -teacher
   +lecturer
   +instructor
 -printed
   +paper
   +magazine
 -web
   +logo
   +webpage
 -assistant
   +ta
 -asist
   +recitation
   +grading
     -assigngrading
 -teach
   +lecturing
   +advice
 -academic
   +advisor
```

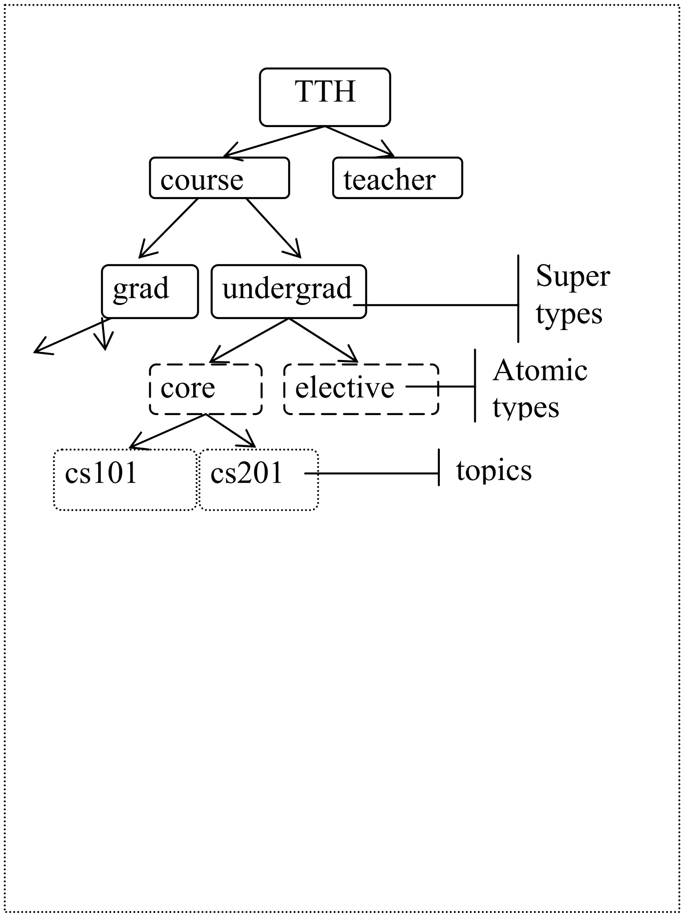


Figure 3.2 TTH

```
-ATH
  -asist
    -recitation
      +five
  -grading
    -assigngrading
      +uc
  -teach
    -lecturing
      +seven
    -six
    -advice
      +two
```

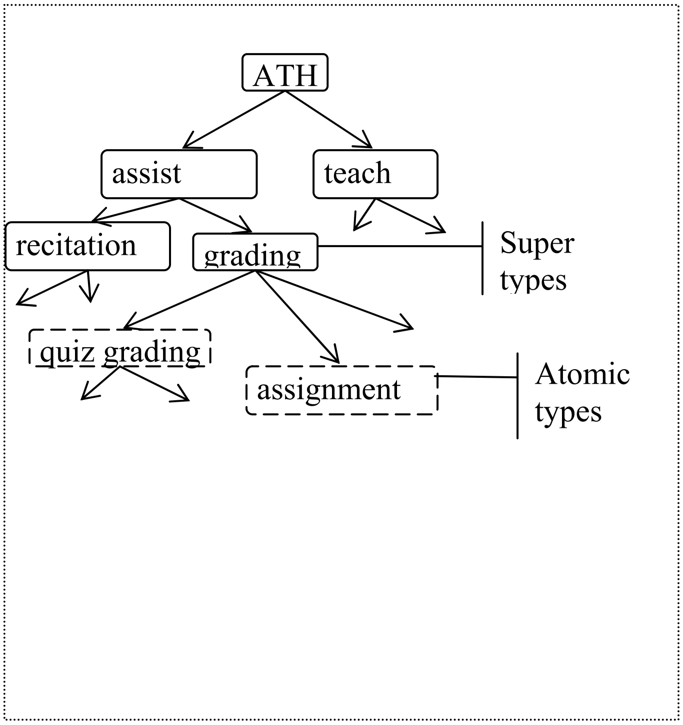


Figure 3.3 ATH

We see the RTH of the Figure 3.1.

```
-RTH
  -course
  -asistant
    +ta
  -teacher
    +lecturer
  -academic
    +advisor
```

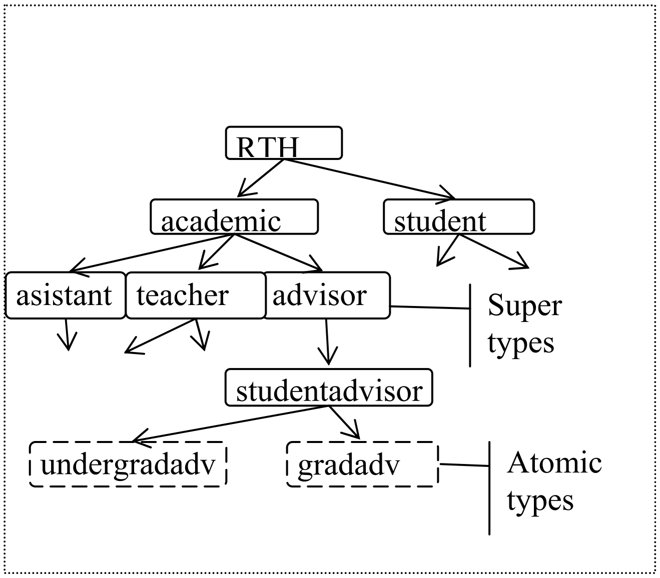


Figure 3.4 RTH

```
-OTH
  -web
    +webpage
    +logo
  -printed
    +paper
    +magazine
```

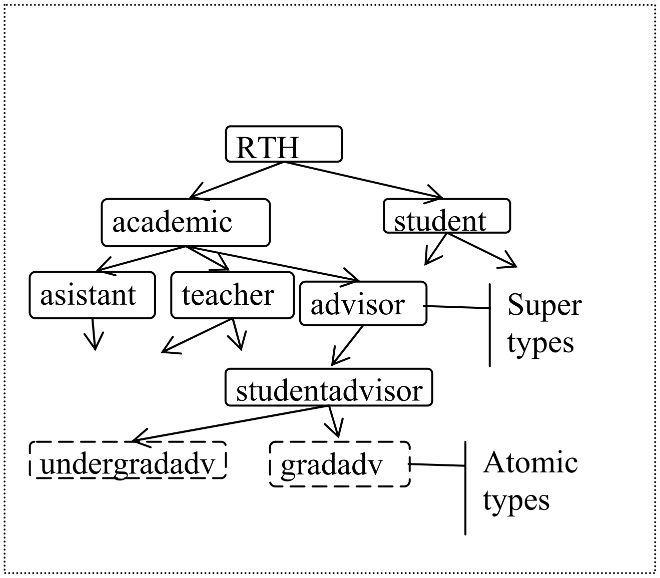


Figure 3.5 OTH

3.3 OBJECT-LEVEL HIERARCHIES

Topic-map level hierarchies do not provide all related entities for a given specific object. This information can be represented using object level hierarchies which are computed from the global hierarchies. A procedure that computes the topic object-level ATH given below. Other object level hierarchies are computed in a similar manner from their respective global level hierarchies.

```
Procedure CreateTopicLevelATH
Input: topic map TM, topic a
Output: a topic type hierarchy
Algorithm:
Let $AT_a \leftarrow \{\}$
Let F be the set of associations incident upon topic a.
Let S be the set of association types in F.
For each assoc type t in S
    Let Super(t) is the set of super assoc types of t.
    Let $AT_a \leftarrow AT_a$ U Super(t)
    For each assoc type f in Super(t)
        Add an edge <t, f> to E
    Delete t from S
Add Super(t) to S
```

Figure 3.6 Create Topic Level ATH Algorithm

Computation starts with computing the set F of associations the topic participates. Next the set S of association types in F is computed from the topic map level ATH in the topic map which can be generated by traversing up to the root of the ATH for each type in S. The topic level ATH, RTH and OTHs for topic "Jane" are given in Figure 3.7 below.

```
-ATH
   -school
     -teach
        -jane
```

```
-TTH
  -person
    -student
       -ta
         -jane
```

```
-OTH
   -web
      -webpage
        -jane
```

```
-RTH
  -teacher
     -lecturer
       -jane
```

Figure 3.7 Sample TTH, ATH, RTH, OTH

```
-jane
   -ATH
     -school
       -teach
   -RTH
     -teacher
       -lecturer
   -OTH
      -web
        -webpage
```

Figure 3.8 Object Level Hierarchy

The object level hierarchies within global hierarchies are depicted in the following figure.

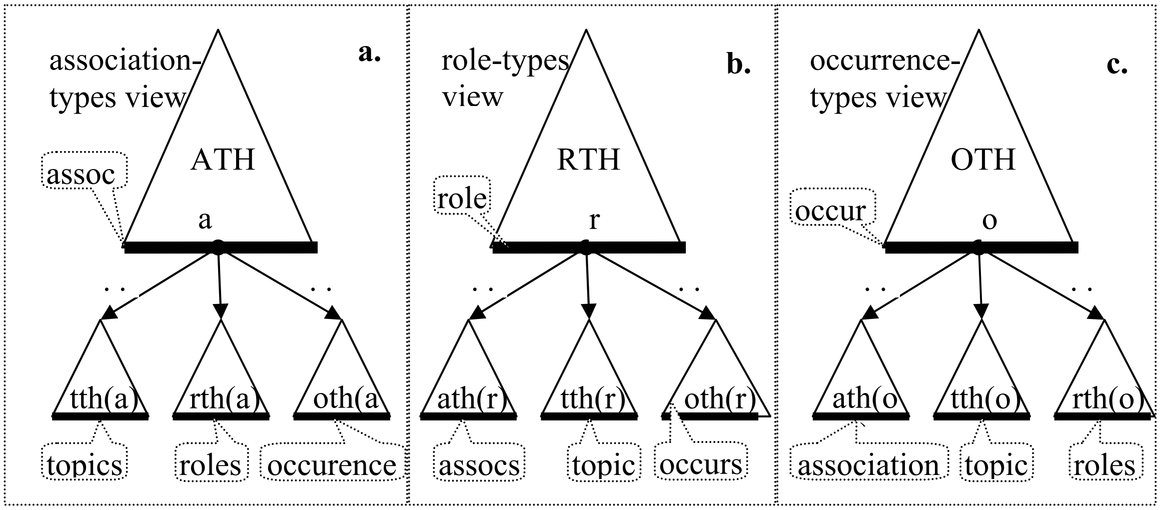


Figure 3.9 Object Level Hierarchies

3.4 TYPE-LEVEL HIERARCHIES

Similar to object level hierarchies, all related entities of a given type in a global type hierarchy should be and can be represented the combination of respective object level hierarchies in the sub-tree of that specific type. The type level hierarchies are computed from object level hierarchies in a similar manner to the procedure given for the object level hierarchy computation. Type level hierarchies are defined for each kind of entities and shown in the following figure conceptually.

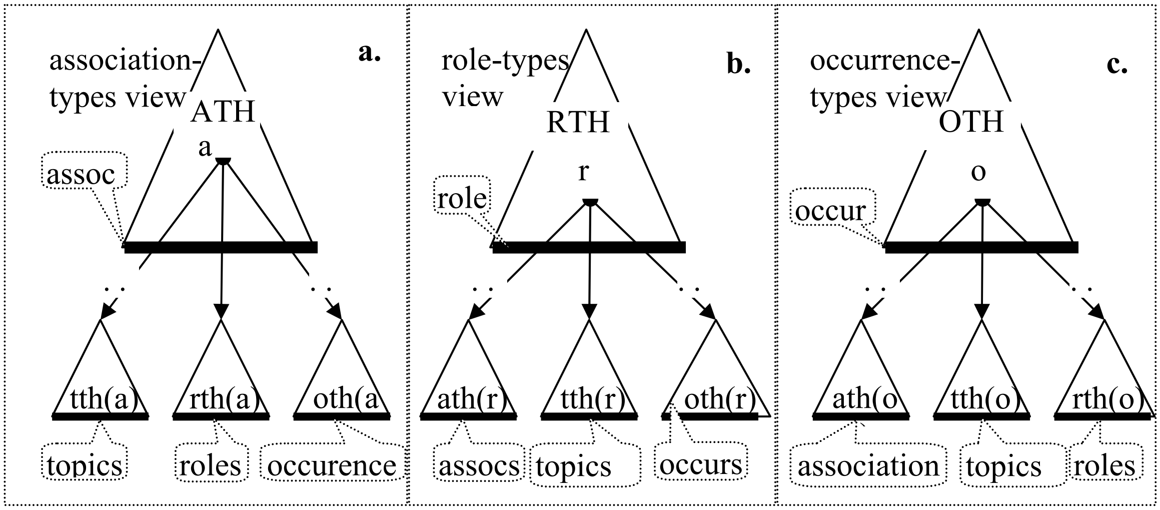


Figure 3.10 Type Level Hierarchies

CHAPTER 4
IMPLEMENTATION

In the study, our aim is to make topic map exploration more comprehensible and easier. In the topic map document entities have hierarchies. But we cannot see it from the xtm syntax. We have prepared a prototype textual exploration tool for topic map using the hierarchies.

We developed the tool in JAVA using Eclipse 3.0 environment. We used SAX parser in *xercesImpl* library to parse the document.

4.1 Data Structure

In the implementation we defined 2 main classes: topic and associations and represented roles and occurences are topics for ease of implementation. The data structure is created by parsing the document and populating the topic and association classes. For each topic object, there a kind attribute indicating the actual type of topic (occurrence, role, topic.), a set vectors of related entities. For example for topic entities the 3 vectors are

- A vector of roles played
- A vector of associations participated
- A vector of occurences

Similarly for other kinds of entities, 3 vectors are defined and maintained. A number of methods for constructing the hierarchies are defined on these classes not given here, but can be find in the code.

4.2 Topic Map Navigation Tool

In our software, we designed a GUI as shown in Figure 4.1. In our GUI, we have four windows to show each hierarchy. At the beginning, user is supposed to open the xml topic map file using the *open* command in the *file* menu.

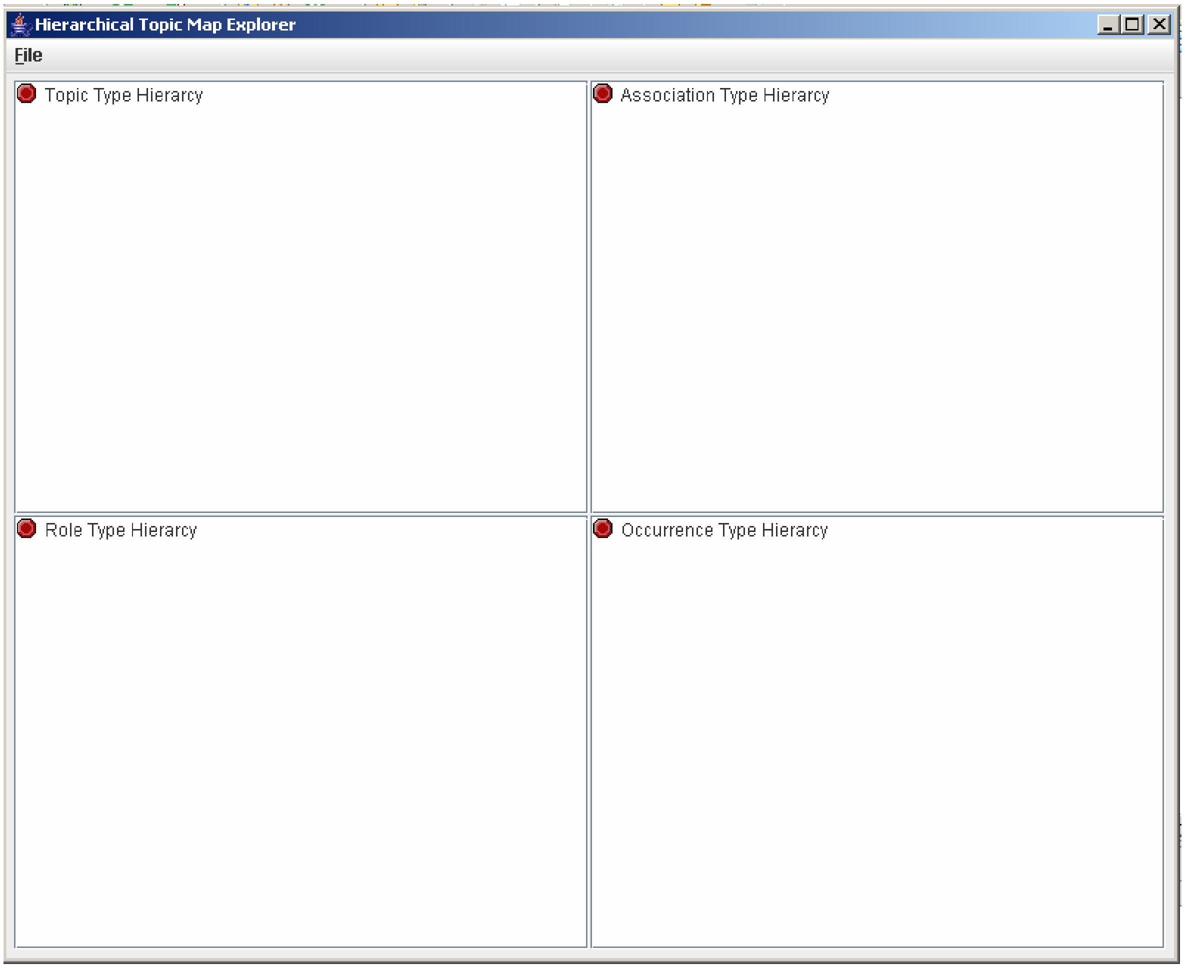
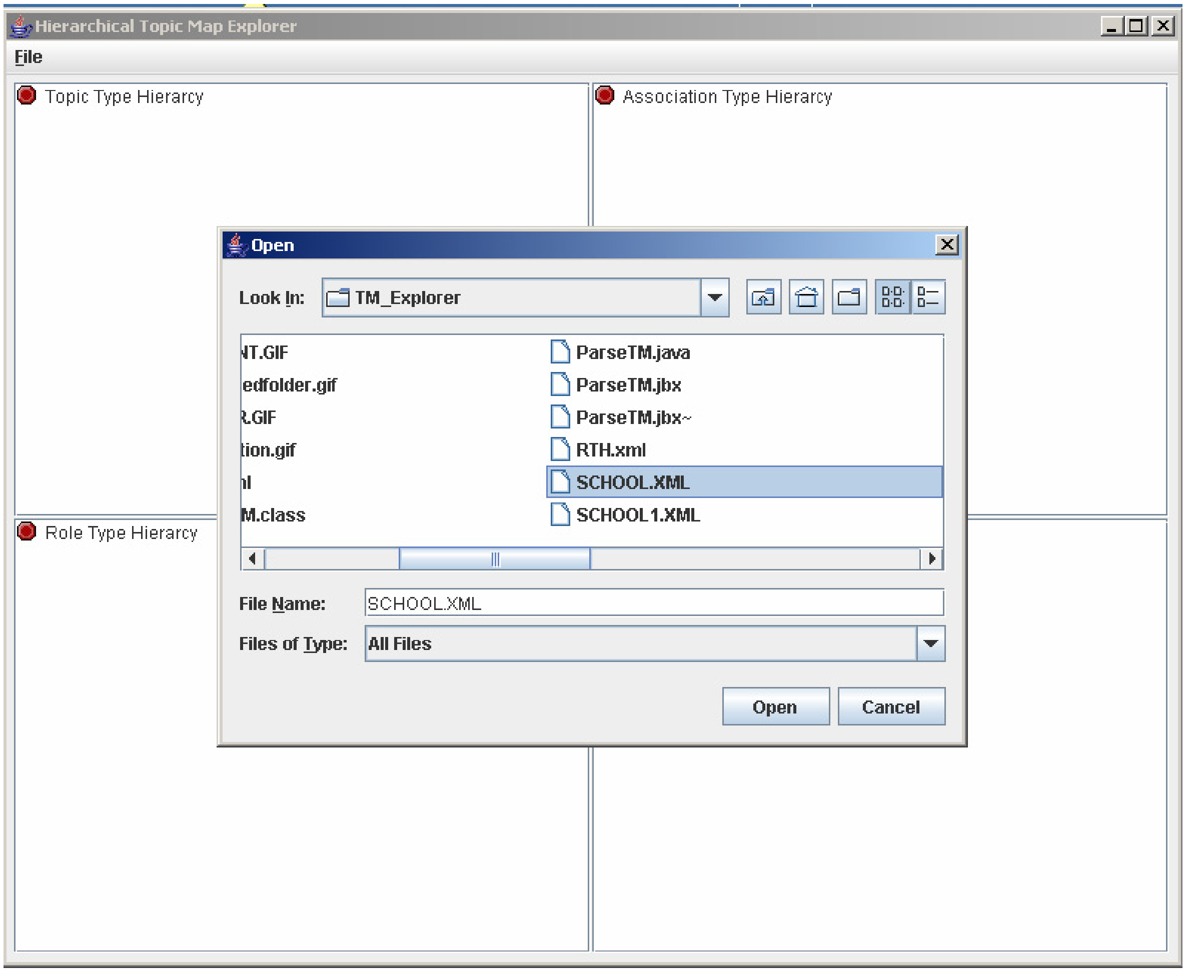


Figure 4.1 Main Appearance

Then user is required to select the topic map document file as shown in Figure 4.2.



Look In: TM_Explorer

File Name: SCHOOL.XML

Files of Type: All Files

Figure 4.2 Parse Topic Map Document

When user selects the topic map document file, software parses the topic map. Then each type hierarchy is listed in the corresponding window as shown in Figure 4.3.

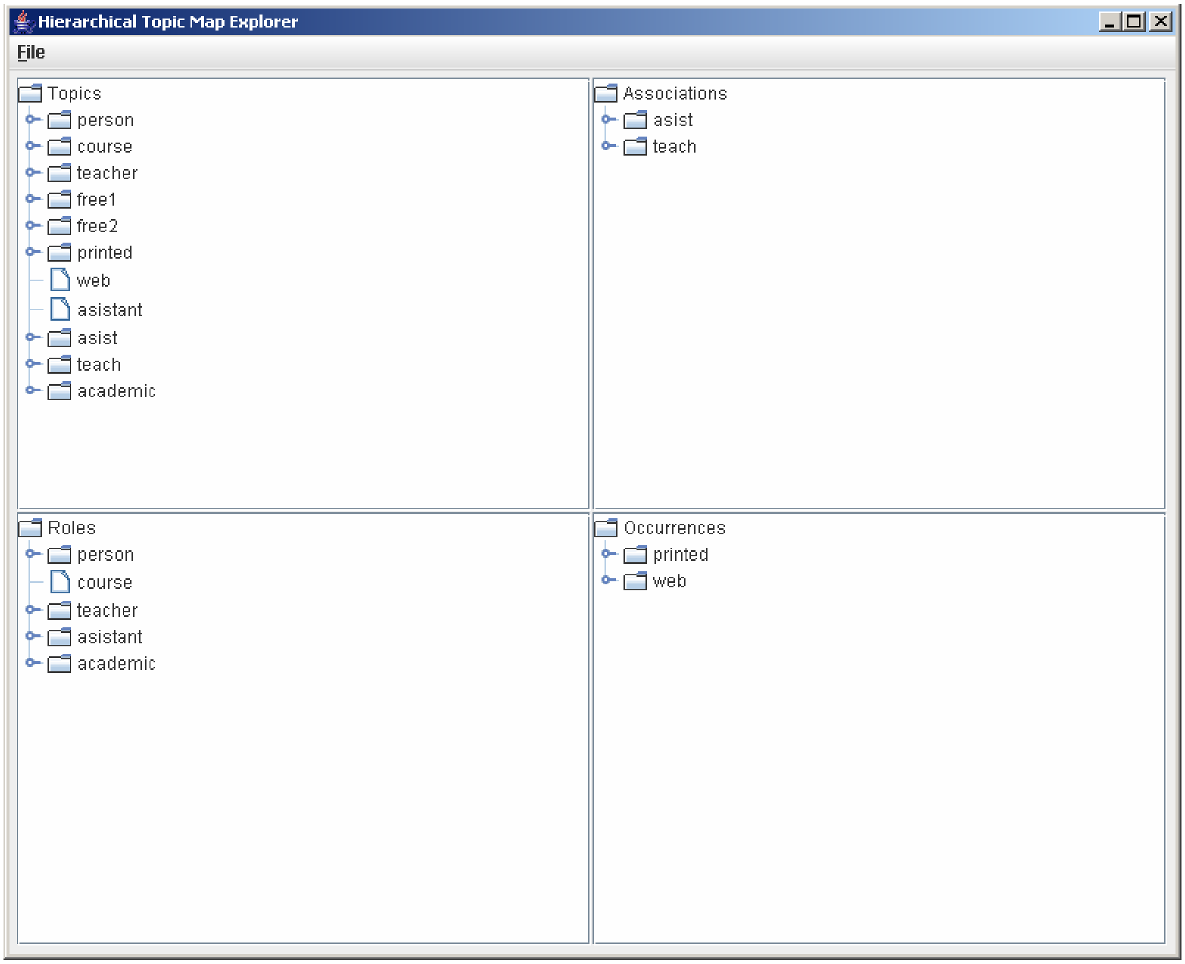


Figure 4.3 Four Type Hierarchies

As shown in Figure 4.4 and Figure 4.5, user can expand and see the details of topics, associations, roles and occurrences in the corresponding type hierarchy window. The "List of nodes for local associations/roles/occurrences of ...." statement is used for listing the nodes in a hierarchy for a given node. It is not required and used for extra precaution to prevent confusion. (It may actually be omitted from output). If the list is empty; this means a local hierarchy doesn't exist.

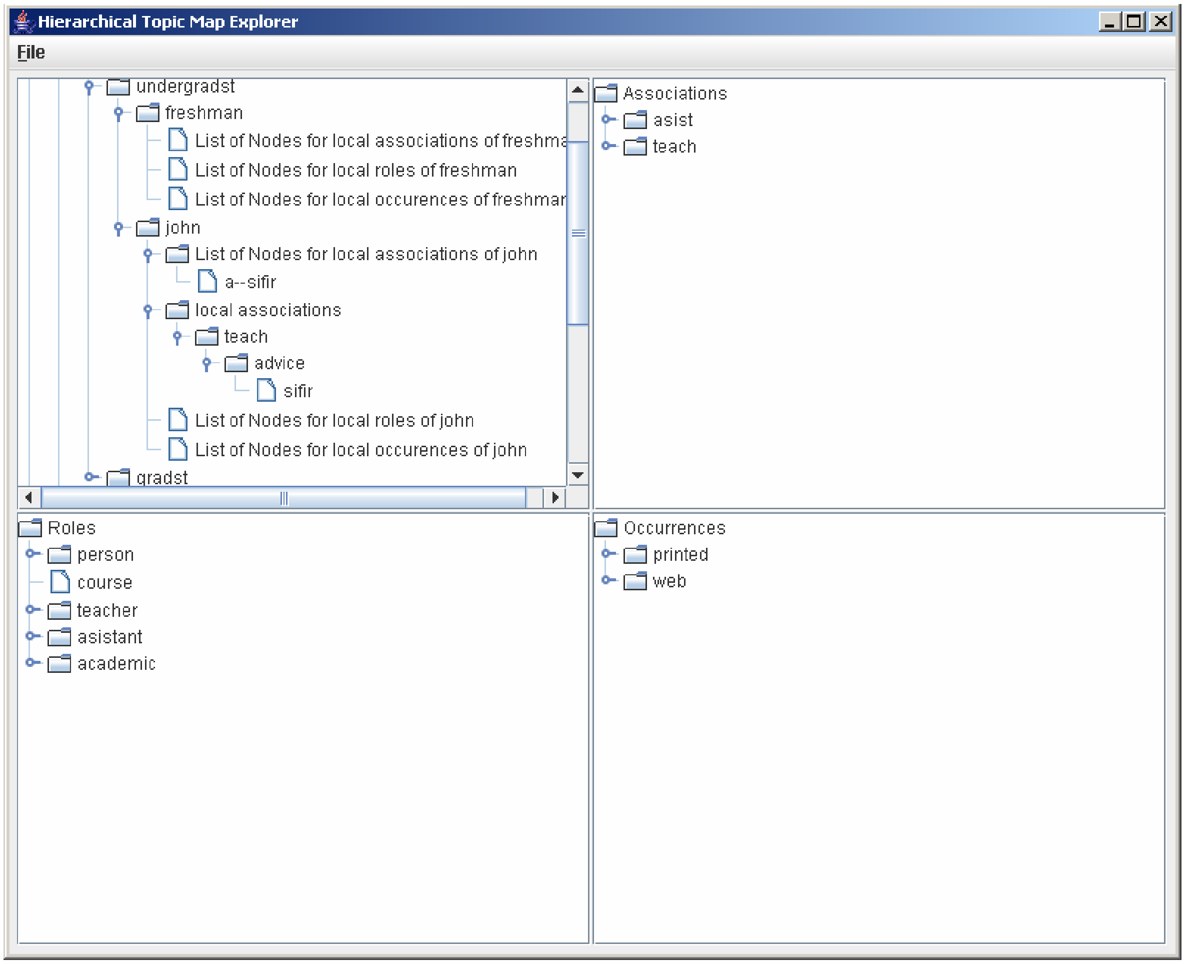

Hierarchical Topic Map Explorer
File
undergradst
freshman
List of Nodes for local associations of freshma
List of Nodes for local roles of freshman
List of Nodes for local occurences of freshmar
john
List of Nodes for local associations of john
a--sifir
local associations
teach
advice
sifir
List of Nodes for local roles of john
List of Nodes for local occurences of john
gradst
Associations
asist
teach
Roles
person
course
teacher
asistant
academic
Occurrences
printed
web

Figure 4.4 Local Hierarchy

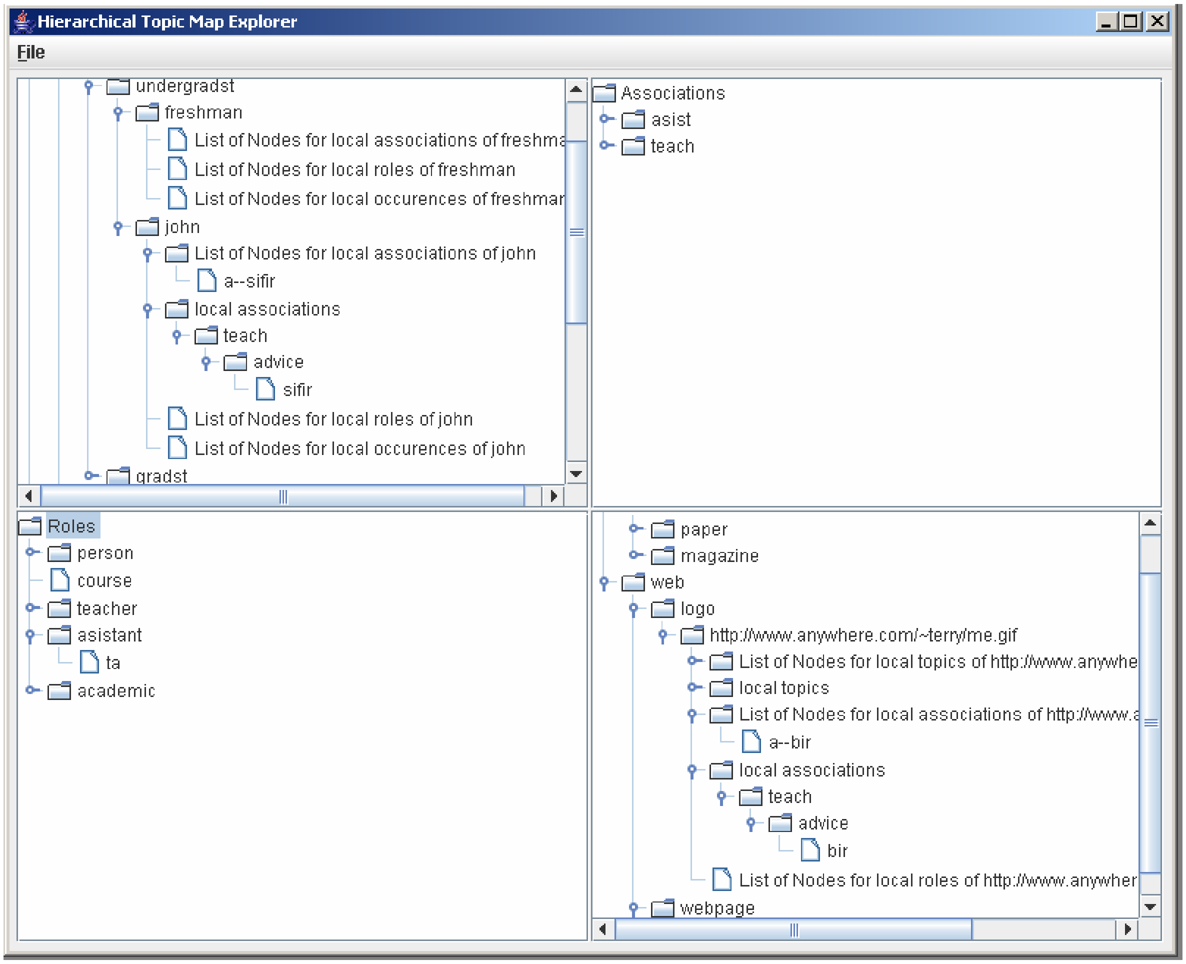

Hierarchical Topic Map Explorer
File
undergradst
freshman
List of Nodes for local associations of freshma
List of Nodes for local roles of freshman
List of Nodes for local occurences of freshmar
john
List of Nodes for local associations of john
a--sifir
local associations
teach
advice
sifir
List of Nodes for local roles of john
List of Nodes for local occurences of john
gradst
Associations
asist
teach
Roles
person
course
teacher
asistant
ta
academic
paper
magazine
web
logo
http://www.anywhere.com/~terry/me.gif
List of Nodes for local topics of http://www.anywhe
local topics
List of Nodes for local associations of http://www.a
a--bir
local associations
teach
advice
bir
List of Nodes for local roles of http://www.anywher
webpage

Figure 4.5 Local Hierarchy

CHAPTER 5
CONCLUSION

Navigating large topic maps may and can be achieved by organizing the entities in the topics maps in separate type hierarchies. Users can choose and switch between these hierarchies for focusing down in more specific entities in topic map. In addition to the hierarchies at global (topic map) level, hierarchies at type or intermediate levels and hierarchies at object or local levels are provided to for a representation of relationship in compact and clear manner. The hierarchies are created for each entity type (topics, associations, roles and occurrences) shown in a separate window within the tool that is implemented.

While the tool is able to demonstrate the ideas presented in this study, it is quite limited in capabilities such as querying, synchronizations among different hierarchies as user navigates down further in one hierarchy.

Recently there has been ongoing work in developing technologies such as semantic web and ontologies to provide a meta level above the huge amount of data on the web. Semantic web is envisioned to be a meta layer on top a data layer (the web pages in the www). Ontologies are one of the important formalisms to represent semantic webs. There are many similarities between topic maps and ontologies which are presented in some well known papers in the literature. Both are based on type or concept hierarchies. However there are more advanced features in ontologies than topic maps. Nevertheless the methodology used here in this study on topic map exploration can be for the most part applied to ontologies in a straightforward manner.

REFERENCES

Biezunski, M., Newcomb, S., Specializing Occurrences in Topic Maps by Association Template Subclassing, In Proceedings of Extreme Markup 2001, Montréal, August 2001, http://www.idealliance.org/papers/extreme02/html/2001/Biezunski01/EML2001Biezunski01.html

Falkovych, K., Sabou, M., Stuckenschmidt, H., UML for the Semantic Web: Transformation-Based Approaches, in: B. Omelayenko, and M. Klein, ed., Knowledge Transformation for the Semantic Web, Frontiers in Artificial Intelligence and Applications, Vol.95 (IOS Press, 2003) pp. 92-106.

Freese, E., So Why Aren't Topic Maps Ruling the World?, In Proceedings of Extreme Markup Languages 2002, August 2002. http://www.idealliance.org/papers/extreme03/html/2002/Freese01/EML2002Freese01.html.

Freese, E., Taking Topic Maps to the Nth Dimension, In Proceedings of Extreme Markup Languages 2003, August 2003, http://www.mulberrytech.com/Extreme/Proceedings/html/2003/Freese01/EML2003Freese01.html

Garshol, L. M., *Topic maps, RDF, DAML, OIL: A comparison*, In Proceedings of XML 2001, IDEAlliance, http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html

Topic maps and graphical structures, http://www.xm.co.nz/op/tm.htm

Garshol, L. M., What Are Topic Maps, 2002, http://www.xml.com/pub/a/2002/09/11/topicmaps.html.

Garshol, L. M., Metadata? Thesauri? Taxonomies? Topic Maps? Making Sense of It All, 2004, http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html

Granmo, G. O., Creating Semantically Valid Topic Maps, In Proceedings of XML Europe 2000, Paris, 12-16 June 2000, http://www.ontopia.net/topicmaps/materials/tm-schemas-paper.pdf

Kal Ahmed, Topic Maps for Repositories, In Proceedings of XML Europe 2000, Paris, June 2000, http://www.gca.org/papers/xmleurope2000/papers/s29-04.html

Kal Ahmed, Developing a Topic Map Programming Model, In Proceedings of XML 2001, Florida, USA, December 9-14, 2001, http://www.idealliance.org/papers/xml2001/papers/html/05-04-03.html

Kalyanpur, A., Pastor, D. J., Battle, S., Podget, J., Automatic Mapping of OWL Ontologies into Java, The Sixteenth International Conference On Software Engineering and Knowledge Engineering, June 20-24, Canada, 2004, http://www.mindswap.org/~aditkal/www2004_OWL2Java.pdf

Ksiezyk, R, Trying not to get lost with a Topic Map, XML Europe '99, April 16-30, Granada, 1999, http://www.infoloom.com/gcaconfs/WEB/granada99/ksi.htm

Ksiezyk, R, Answer Is Just a Question [of Matching Topic Maps], In Proceedings of XML Europe 2000, June 12-16, Paris, 2000, http://www.gca.org/papers/xmleurope2000/papers/s22-03.html

Kunz, C., Botsch, V., Visual Representation and Contextualization of Search Results – List and Matrix Browser, In Proceedings of Dublin Core´02, Florence, Italy, 2002, http://www.bncf.net/dc2002/program/ft/poster10.pdf

Le Grand, B., Soto, M., Information Management - Topic Maps Visualization, In Proceedings of XML Europe 2000, June 12-16, Paris, 2000, http://www.gca.org/papers/xmleurope2000/papers/s29-03.html

Le Grand, B., Soto, M., Visualization of Semantic Web: Topic Maps Visualisation, Information Visualisation, IV 2002, http://www-rp.lip6.fr/site_npa/site_rp/_publications/244-LeGrand_TopicMaps.pdf

Le Grand, B., Soto, M., Conceptual Exploration of Topic Maps, Conference Presentation in XML 2000, December 3-8, 2000. http://www-rp.lip6.fr/site_npa/site_rp/_publications/207-XML2000paper.pdf

Lin,X., Soergel, D., Marchionini, G., A Self-organizing Semantic Web For Information Retrieval, in Proceedings of the 14. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR91), Chicago, IL, October 13 - 16 1991, pp. 262-269.

Mumcu K, Topic Map Visualization, Master Thesis at faith University, İstanbul, 2005

Pepper, S., Methods for the Automatic Construction of Topic Maps, 2002, http://www.ontopia.net/topicmaps/materials/autogen-pres.pdf

Pepper, S., The TAO of Topic Maps, In Proceedings of XML Europe 2000, June 2000, http://www.ontopia.net/topicmaps/materials/tao.html

Pepper, S., Euler, Topic Maps and Revolution, In Proceedings of XML Europe '99, Granada, April 1999, http://www.ontopia.net/topicmaps/materials/euler.pdf

Pepper, S., Navigating Haystacks and Discovering Needles, In Sperberg-McQueen, C.M.; Usdin, B.T. (eds): Markup Languages, vol.1, no.4, MIT Press, Cambridge (MA), 1999, http://www.ontopia.net/topicmaps/materials/mlangart.pdf

Protégé (n. d.), http://protege.stanford.edu/

Rath, H. H., Topic Maps Self-control, In Proceedings of Extreme Markup Language, 2000, http://www.gca.org/attend/2000_conferences/Extreme_2000/Papers/Rath/tom-hhr.PDF.

Rath, H. H., Pepper, S., Topic Maps: Introduction and Allegro, Markup Technologies 99, Philadelphia, USA, December 1999, http://www.ontopia.net/topicmaps/materials/allegro.pdf.

Roxborough, T., Sen, A., Graph Clustering Using Multiway Ratio Cut, In Proceedings of Graph Drawing, volume 1353 of Lect. Notes in Cornput. Sci., pages 291-296, Springer-Verlag, New York/Berlin, 1997.

Schmidt, I., Müller, C., A World to Discover: Topic Map for Thomas Mann, In Proceedings of XML Europe 2001, 21-25 May 2001, Internationals Congress Centrum (ICC), Berlin, Germany, http://www.gca.org/papers/xmleurope2001/papers/html/s10-2.html.

Shneiderman, B., The eye have it: A task by data type taxonomy for information visualizations, In Proceedings of IEEE Visual Languages , pages 336-343, Boulder, CO, Sept 1996, http://citeseer.ist.psu.edu/shneiderman96eyes.html

Shrimp (n. d.), http://www.thechiselgroup.org/shrimp

Smolnik, S., Erdmann, Visual Navigation of Distributed Knowledge Structures in Groupware-based Organizational Memories, In Proceedings of Sixth International Conference on Information Visualisation (IV'02), London, England, July, 10 –12, 2002, p. 353-360.

Spoerri, A., InfoCrystal: A Visual Tool for Information Retrieval and Management, In Proceedings of the second international conference on Information and knowledge management , Washington, D.C., November 1993, pp. 11--20.

Techquila (n. d.), A Practical Introduction to Topic Maps, http://www.techquila.com/practical_intro.html

Techquila (n. d.), Topic Map Design Patterns for Information Architecture, http://www.techquila.com/tmsinia.html

Wrightson, A., Topic Maps and Knowledge Representation, February 2001, http://www.ontopia.net/topicmaps/materials/kr-tm.html

XML Topic Maps (XTM) 1.0, 2001, http://www.topicmaps.org/xtm/1.0/ .

APPENDIX A

XTM 1.0 DOCUMENT TYPE DECLARATION

```
<!-- .......................................................... -->
<!-- XML Topic Map DTD  ........................................ -->
<!-- topicMap: Topic Map document element ....................... -->
<!ELEMENT topicMap ( topic | association | mergeMap )*  >
<!ATTLIST topicMap
  id            ID       #IMPLIED
  xmlns         CDATA    #FIXED 'http://www.topicmaps.org/xtm/1.0/'
  xmlns:xlink   CDATA    #FIXED 'http://www.w3.org/1999/xlink'
  xml:base      CDATA    #IMPLIED  >

<!-- topic: Topic element ...................................... -->
<!ELEMENT topic ( instanceOf*, subjectIdentity?, ( baseName | occurrence )* )  >
<!ATTLIST topic     id            ID       #REQUIRED  >

<!-- instanceOf: Points To a Topic representing a class .......... -->
<!ELEMENT instanceOf ( topicRef | subjectIndicatorRef ) >
<!ATTLIST instanceOf     id            ID       #IMPLIED  >

<!-- subjectIdentity: Subject reified by Topic .................. -->
<!ELEMENT subjectIdentity ( resourceRef?, ( topicRef | subjectIndicatorRef )* ) >
<!ATTLIST subjectIdentity     id            ID       #IMPLIED  >

<!-- topicRef: Reference to a Topic element ..................... -->
<!ELEMENT topicRef  EMPTY >
<!ATTLIST topicRef
  id            ID       #IMPLIED
  xlink:type     NMTOKEN   #FIXED 'simple'
  xlink:href     CDATA     #REQUIRED  >

<!-- subjectIndicatorRef: Reference to a Subject Indicator ....... -->
<!ELEMENT subjectIndicatorRef  EMPTY >
<!ATTLIST subjectIndicatorRef
  id            ID       #IMPLIED
  xlink:type     NMTOKEN   #FIXED 'simple'
  xlink:href     CDATA     #REQUIRED  >

<!-- baseName: Base Name of a Topic ............................. -->
```

```
<!ELEMENT baseName  ( scope?, baseNameString, variant* ) >
<!ATTLIST baseName     id              ID        #IMPLIED  >

<!-- baseNameString: Base Name String container ................. -->
<!ELEMENT baseNameString  ( #PCDATA ) >
<!ATTLIST baseNameString     id              ID        #IMPLIED  >

<!-- variant: Alternate forms of Base Name ...................... -->
<!ELEMENT variant  ( parameters, variantName?, variant* ) >
<!ATTLIST variant     id              ID        #IMPLIED  >

<!-- variantName: Container for Variant Name .................... -->
<!ELEMENT variantName  ( resourceRef | resourceData ) >
<!ATTLIST variantName     id              ID        #IMPLIED  >

<!-- parameters: Processing context for Variant ................. -->
<!ELEMENT parameters  ( topicRef | subjectIndicatorRef )+ >
<!ATTLIST parameters     id              ID        #IMPLIED  >

<!-- occurrence: Resources regarded as an Occurrence ............ -->
<!ELEMENT occurrence ( instanceOf?, scope?, ( resourceRef | resourceData ) )  >
<!ATTLIST occurrence     id              ID        #IMPLIED  >

<!-- resourceRef: Reference to a Resource ....................... -->
<!ELEMENT resourceRef  EMPTY >
<!ATTLIST resourceRef
  id              ID        #IMPLIED
  xlink:type      NMTOKEN   #FIXED 'simple'
  xlink:href      CDATA     #REQUIRED  >

<!-- resourceData: Container for Resource Data .................. -->
<!ELEMENT resourceData  ( #PCDATA ) >
<!ATTLIST resourceData     id              ID        #IMPLIED  >

<!-- association: Topic Association  ............................ -->
<!ELEMENT association ( instanceOf?, scope?, member+ ) >
<!ATTLIST association     id              ID        #IMPLIED  >

<!-- member: Member in Topic Association ........................ -->
<!ELEMENT member ( roleSpec?, ( topicRef | resourceRef | subjectIndicatorRef )* ) >
<!ATTLIST member     id              ID        #IMPLIED  >

<!-- roleSpec: Points to a Topic serving as an Association Role .. -->
<!ELEMENT roleSpec  ( topicRef | subjectIndicatorRef ) >
<!ATTLIST roleSpec     id              ID        #IMPLIED  >
```

```
<!-- scope: Reference to Topic(s) that comprise the Scope ........ -->
<!ELEMENT scope  ( topicRef  | resourceRef | subjectIndicatorRef )+ >
<!ATTLIST scope    id            ID       #IMPLIED  >

<!-- mergeMap: Merge with another Topic Map ..................... -->
<!ELEMENT mergeMap  ( topicRef | resourceRef | subjectIndicatorRef )* >
<!ATTLIST mergeMap
  id             ID       #IMPLIED
  xlink:type      NMTOKEN   #FIXED 'simple'
  xlink:href      CDATA     #REQUIRED  >

<!-- end of XML Topic Map (XTM) 1.0 DTD -->
```

APPENDIX B

SCHOOL.XTM DOCUMENT

```
<topicmap xmlns:xlink="http://www.w3.org/1999/xlink" commonThemes="mary">
   <topic id = "person" />
   <topic id = "course" />
   <topic id = "teacher" />
   <topic id = "free1"/>
   <topic id = "free2"/>
   <topic id = "student" instanceOf = "person" />
   <topic id = "undergradst" instanceOf = "student" />
   <topic id = "freshman" instanceOf = "undergradst" />
   <topic id = "gradst" instanceOf = "student" />
   <topic id = "instructor" instanceOf = "person teacher" />
   <topic id = "undergradcourse" instanceOf = "course" />
   <topic id = "core" instanceOf = "undergradcourse" />
   <topic id = "elective" instanceOf = "undergradcourse" />
   <topic id = "gradcourse" instanceOf = "course" />
   <topic id = "printed" />
   <topic id = "web" />
   <topic id = "book" instanceOf = "printed" />
   <topic id = "paper" instanceOf = "printed" />
   <topic id = "magazine" instanceOf = "printed" />
   <topic id = "logo" instanceOf = "web" />
   <topic id = "webpage" instanceOf = "web" />
   <topic id = "john" instanceOf = "undergradst" />
   <topic id = "joe" instanceOf = "gradst"/>
   <topic id = "sue" instanceOf = "gradst" />
   <topic id = "mary" instanceOf = "instructor" />
   <topic id = "terry" instanceOf = "instructor" >
     <occurs type = "logo" href="http://www.anywhere.com/~terry/me.gif"
/>
     <occurs type = "magazine" href = "http://www.magazines.com/~terry"
/>
   </topic>
   <topic id = "jane" instanceOf = "instructor">
     <occurs type = "paper"
href="http://www.somewhere.com/~jane/papaers.html" />
     <occurs type = "webpage" href = "http://www.somewhere.com/~jane" />
   </topic>
   <topic id = "cs601" instanceOf = "gradcourse" />
   <topic id = "cs101" instanceOf = "core" />
   <topic id = "cs201" instanceOf = "core" />
   <topic id = "cs203" instanceOf = "core" />
   <topic id = "asistant" />
   <topic id = "ta" instanceOf = "asistant" />
   <topic id = "asist" />
   <topic id = "recitation" instanceOf = "asist" />
   <topic id = "grading" instanceOf = "asist" />
   <topic id = "assigngrading" instanceOf = "grading" />
   <topic id = "quizgrading" instanceOf = "grading" />
   <topic id = "midtermgrading" instanceOf = "grading" />
   <topic id = "teach" />
   <topic id = "lecturing" instanceOf = "teach" />
   <topic id = "advice" instanceOf = "teach" />
   <topic id = "doofficehour" instanceOf = "teach" />
   <topic id = "lecturer" instanceOf = "teacher" />
   <topic id = "academic" />
   <topic id = "advisor" instanceOf = "academic" />
   <topic id = "studentadvisor" instanceOf = "advisor" />
   <topic id = "undergradadvisor" instanceOf = "studentadvisor" />
   <topic id = "gradadvisor" instanceOf = "studentadvisor" />
   <assoc id ="zero" instanceOf = "advice" >
     <assocrl xlink:href="#mary" role="undergradadvisor"/>
     <assocrl xlink:href="#john" role="undergradst"/>
   </assoc>
   <assoc id = "one" instanceOf = "advice" >
     <assocrl xlink:href="#terry" role="gradadvisor"/>
     <assocrl xlink:href="#joe" role="gradst"/>
   </assoc>
   <assoc id = "two" instanceOf = "advice" >
     <assocrl xlink:href="#jane" role="advisor"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "three" instanceOf = "assigngrading" >
     <assocrl xlink:href="#sue" role="ta"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "four" instanceOf = "quizgrading" >
     <assocrl xlink:href="#sue" role="ta"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "five" instanceOf = "recitation" >
     <assocrl xlink:href="#sue" role="ta"/>
     <assocrl xlink:href="#cs101" role="course"/>
   </assoc>
   <assoc id = "six" instanceOf = "teach" >
     <assocrl xlink:href="#jane" role="teacher"/>
     <assocrl xlink:href="#cs201" role="course"/>
   </assoc>
      <assoc id = "seven" instanceOf = "lecturing" >
     <assocrl xlink:href="#mary" role="lecturer"/>
     <assocrl xlink:href="#cs601" role="course"/>
   </assoc>
   <assoc id = "eight" instanceOf = "midtermgrading" >
     <assocrl xlink:href="#jane" role="instructor"/>
     <assocrl xlink:href="#cs601" role="course"/>
   </assoc>
</topicmap>
```

APPENDIX C

SOFTWARE CODES

```
Topic.java

import java.util.*;

public class Topic {
  String id;
  char nodetype;  // '' innernode/type, 't'  topic, 'o' occur, 'a' assoc, 'r' role
  boolean flag = false;  // used as an aid in some methods
  boolean isTopicType = false, isAssocType = false, isRoleType = false, isOccurType = false;
  Vector supertypes = new Vector();
  Vector subtypes = new Vector();
  boolean messages  = true; // if true display messages for tracing program

  public Topic(){}
  public Topic(String id) {this.id = id;}

  public boolean equals (Topic o){
      System.out.println("--"+this.id);
      return (id.equals(o.id));
  }
  public Topic(String id, Vector types) {
    this.id = id;
    this.supertypes = types;
  }

  public String getID(){     return this.id;   }
  public void setId(String newid){     this.id = newid;  }
  public String getDisplayName() {
      String aux= nodetype+"-";
      if (this.isTopicType) aux += "t";
      if (this.isOccurType) aux += "o";
      if (this.isAssocType) aux += "a";
      if (this.isRoleType)  aux += "r";
      aux += "-"+id;
      return aux;
  }
  public Vector getSuperTypes(){ return this.supertypes; }
  public Vector getSubTypes()  { return this.subtypes;   }
  public void addSuperType(Topic type) {this.supertypes.addElement(type);  }
  public void addSubType(Topic type)   {this.subtypes.addElement(type); }
  public void update_parents(char type){

      if (this.nodetypeOK(type)){ // no need to update, parents must be already set
            if (messages) System.out.println(this.getID()+" is not of right type
returning!");
            return;
      }
      this.setNodeType(type);
      if (messages) System.out.println(this.getID()+": changed type to "+type);
      //System.out.println("> "+this.getDisplayName());
      Vector parents = this.getSuperTypes();
      if (! parents.isEmpty()){
            if (messages) System.out.println(this.getID()+": updating parents ");
            for (int i=0; i < parents.size(); i++){
                  Topic t = (Topic)parents.elementAt(i);
                  t.update_parents(type);
            }
      }else
            if (messages) System.out.println(this.getID()+" no parents ");
  }

  public boolean nodeIsOf(char type){
      return (type==nodetype);
  }
  public boolean nodetypeOK(char type){ //change name to nodeTypeIsOF
      // for the internal nodes
      // checks if node is of given 'type'
      return ((type=='t' && this.isTopicType) ||
                  (type=='o' && this.isOccurType) ||
                  (type=='a' && this.isAssocType) ||
                  (type=='r' && this.isRoleType));
  }
  public void setNodeType(char type) {
      if (type=='t') this.isTopicType = true;
      if (type=='o') this.isOccurType = true;
      if (type=='a') this.isAssocType = true;
      if (type=='r') this.isRoleType  = true;
  }
  public boolean isleafnode(){
      return (this.nodetype=='t' ||
                  this.nodetype=='o' ||
                  this.nodetype=='a' ||
                  this.nodetype=='r');
  }
  // checks if this topic is included in the topics vector
  // returns the location of the topic in the vector, -1 if not found
  public int containedin (Vector topics){
      for (int i=0; i<topics.size(); i++)
            if (this.id.equals(((Topic)topics.elementAt(i)).id))
                  return i;
      return -1;
  }
}
```

Association.java

```
import java.util.*;

public class Association extends Topic {

  Vector roles = new Vector();

  public Association(String id) {
    this.id = id;
    this.nodetype = 'a';
  }
  public void addMember(Topic member, Topic role) {
    this.subtypes.add(member);
    this.roles.add(role);
  }
  public Vector getMembers(){ return this.getSubTypes();}
  public Vector getRoles()    {     return this.roles;      }
}
```

ParseTM.java

```
import java.io.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import java.util.*;

public class ParseTM extends DefaultHandler {
      Vector topics = new Vector(); // all topics in the topic map including <topicmap>
element
      /*
       * topics holds a type hierarchy of everything in the topic map.
       * the inner nodes (where nodetype is empty) of this hiearchy rooted at
t=topics.firstelement() are types
       * the leaf nodes (where nodetype is t, o, a, or r) of this hieарcy are topics,
       * associations, occurences or roles
       * the global (topic map) level hierachies are created from this data structure
       */
      Topic tm = null; // holds <topicmap> topic that are the superclass of all topics, the
root
      Topic currentTopic;
      Association currentAssoc;
      boolean messages = false; // for printing informative messages to trace program
execution
      public void startDocument() throws SAXException{      }

      public void endDocument() throws SAXException{
            System.out.println();
            printTopicMap((Topic)topics.firstElement(), 0);
      }
      // method called at the start tag of an element
      public void startElement(String uri, String eleName, String raw, Attributes
attributes) throws SAXException{

            if (topics.size()==0 && eleName.equals("topicmap")){ // the outmost element name
must be topicmap
                  tm = new Topic("topicmap");
                  topics.addElement(tm);
            }
            else if (eleName == "topic"){ // this is a base topic (not a topic type)
                  currentTopic = new Topic(attributes.getValue("id"));
                  currentTopic.nodetype = 't'; // make it base topic
                  if (currentTopic.containedin(topics) == -1) // add it to list of topics
                        topics.addElement(currentTopic);
                  else {
                        System.out.println("Syntax error: Cannot redefine a topic! Can't
continue, terminating!");
                        System.exit(0);
                  }
                  if (attributes.getValue("instanceOf") == null) { // if this topic NOT have
supertypes
                        currentTopic.addSuperType(tm);            // then it is an instance
of TOP type
                        tm.addSubType(currentTopic);
                        tm.isTopicType = true;
                  }else {                                                      // if no super
type is specified
                        String tokens[] = attributes.getValue("instanceOf").split(" ");
                        for (int i=0; i < tokens.length; i++){    // for each super type
                              if (messages) System.out.println(currentTopic.getID()+" is
instanceof "+tokens[i]);
                              Topic t = new Topic(tokens[i]);
                              t.isTopicType = true;
                              int aux = t.containedin(topics); // search existing topics so
far
```

```
                              if (aux == -1) {                    // not found, then a new
topic
                                    topics.addElement(t);         // then add as a new topic
                                    t.addSubType(currentTopic);
                                    t.addSuperType(tm);
                                    tm.addSubType(t);
                                    tm.isTopicType = true;
                                    currentTopic.addSuperType(t);
                                    if (messages) System.out.println(tokens[i]+" is not
already a topic!");
                              }else{                                          // already
exists in TM, then update type hiearachy
                                    Topic s = (Topic)topics.elementAt(aux);
                                    s.nodetype = ' ';
                                    s.addSubType(currentTopic);
                                    currentTopic.addSuperType(s);
                                    s.update_parents('t');
                              }
                        }
                  }
            }
            else if (eleName == "assoc") {
                  currentAssoc = new Association(attributes.getValue("id"));
                  if (currentAssoc.containedin(topics) == -1)
                        topics.addElement(currentAssoc);
                  else {
                        System.out.println("Syntax error: Cannot redefine an existing topic
as a new association! Cant continue, terminating!");
                        System.exit(0);
                  }
                  if (attributes.getValue("instanceOf") == null) {
                        currentAssoc.addSuperType(tm);
                        tm.addSubType(currentAssoc);
                        tm.isAssocType = true;
                  }else {                                                     // if no super
type is specified
                        String tokens[] = attributes.getValue("instanceOf").split(" ");
                        for (int i=0; i < tokens.length; i++){    // for each super type
                              if (messages) System.out.println(currentAssoc.getID()+" is
instanceof "+tokens[i]);
                              Topic t = new Topic(tokens[i]);
                              t.isAssocType = true;               // set type of topic
                              int aux = t.containedin(topics); // search existing topics so
far
                              if (aux == -1) {                    // not found
                                    topics.addElement(t);
                                    currentAssoc.addSuperType(t);
                                    t.addSubType(currentAssoc);   // then add t as a new
topic
                                    t.addSuperType(tm);
                                    tm.addSubType(t);
                                    tm.isAssocType = true;
                                    if (messages) System.out.println(tokens[i]+" is not
already a topic!");
                              }else { // else update type hiearachy
                                    Topic s = ((Topic)topics.elementAt(aux));
                                    if (messages) System.out.println("Parent of
"+currentAssoc.getDisplayName()+" is "+s.getDisplayName());
                                    s.nodetype = ' ';
                                    s.addSubType(currentAssoc);
                                    currentAssoc.addSuperType(s);
                                    if (messages) System.out.println("Updating parent of
"+currentAssoc.getDisplayName());
```

```
                                    s.update_parents('a');
                                    if (messages) System.out.println("Finished updating 
parent of "+currentAssoc.getDisplayName());
                              }
                        }
                  }
            }
            else if(eleName == "assocrl"){
                  if (currentAssoc == null) {
                        System.out.println("Syntax error: Assocurl element outside 
association element! Cant continue, terminating!");
                        System.exit(0);
                  }
                  // Member topics are indicated as value of "@xlink:href" or "@href" 
attribute values in the sample document
                  Topic member = null;
                  if(attributes.getValue("xlink:href")!=null)
                        member = new 
Topic(attributes.getValue("xlink:href").substring(1,attributes.getValue("xlink:href").length
()));
                  if(attributes.getValue("href")!=null)
                        member = new Topic(attributes.getValue("href"));
                  if (member == null){
                        System.out.println("Syntax error: Assocurl element doesnot have href 
attribute. Terminating!");
                        System.exit(0);
                  }
                  int aux1 = member.containedin(topics);
                  if (aux1 == -1){
                        System.out.println("Syntax error: href attribute("+member.getID()+") 
in Assocurl refers to a non-existing topic. Terminating!");
                        if (messages) printTopics(topics);
      System.exit(0);
                  }
                  member = (Topic)topics.elementAt(aux1);
                  if (!( member.nodeIsOf('t') || member.nodetypeOK('t'))){ // remember that 
member must already exists
                        member.update_parents('t');
                        if (messages) System.out.println("updating parents of 
"+member.getDisplayName()+ " for "+currentAssoc.getDisplayName());
                  }

                  if (attributes.getValue("role") != null){
                        Topic role = new Topic(attributes.getValue("role"));
                        role.nodetype = 'r';                      // create a base role
                        int aux = role.containedin(topics); // search existing topics so far
                        if (aux == -1) {                          // not found
                              if (messages) System.out.println("Creating a top level role 
"+role.getDisplayName()+ " for "+currentAssoc.getDisplayName());
                              topics.addElement(role);
                              role.addSubType(currentAssoc);
                              role.addSuperType(tm);
                              tm.addSubType(role);
                              tm.isRoleType = true;
                        }else{                                                // else update 
type hiearachy
                              Topic s = ((Topic)topics.elementAt(aux));
                              s.nodetype = ' ';                   // ???
                              s.addSubType(currentAssoc);
                              if (messages) System.out.println("Updating existing role: 
"+s.getDisplayName()+ " for "+currentAssoc.getDisplayName());
                              s.update_parents('r');
                        }
                        currentAssoc.addMember(member, role); //??? data structure is unable
to store role,member
                  } else {
                        System.out.println("Syntax error: Assocurl doesnt have role
attribute. Terminating!");                      System.exit(0);
                  }
            }
            else if(eleName == "occurs"){

                  if (currentTopic == null){
                        System.out.println("Syntax error: Occurrence element must not occur
outside topic element. Terminating!");
                        System.exit(0);
                  }
                  if (attributes.getValue("href")==null ||
attributes.getValue("type")==null){
                        System.out.println("Syntax error: Occurs element doesnt have type or
href attributes. Terminating!");
                        System.exit(0);
                  }
                  // process the occurence
                  Topic o = new Topic(attributes.getValue("href"));
                  o.nodetype = 'o';                   //create a base occurence

                  int aux = o.containedin(topics); // search existing topics so far
                  if (aux == -1) {                    // not found
                        topics.addElement(o);
                  }else{                                          // else update type
hiearachy
                        o = ((Topic)topics.elementAt(aux));
                        o.nodetype = ' ';
                        o.update_parents('o');
                  }
                  o.addSubType(currentTopic);   // means occurence o occurs in topic
currentTopic
                  currentTopic.addSubType(o); // means o is an occur of currentTopic

                  // process the type of occurence
                  Topic ot = new Topic(attributes.getValue("type"));
                  ot.isOccurType = true;                    //create a base occur type
                  ot.addSubType(o);
                  int aux1 = ot.containedin(topics);  // search existing topics so far
                  if (aux1 == -1){                          // not found
                        topics.addElement(ot);
                        ot.addSuperType(tm);
                        tm.addSubType(ot);
                        tm.isOccurType = true;
                  }else{                                          // else update type
hiearachy
                        ot = ((Topic)topics.elementAt(aux1));
                        ot.nodetype = ' ';
                        ot.addSubType(o);
                        ot.update_parents('o');
                  }

            }//end elseif for eleName=occurs
      }/*end of startelement function*/

      /*method called at the end tag of an element*/
      public void endElement( String uri, String eleName, String raw ) throws SAXException{
            if(eleName == "assoc")  currentAssoc = null;
            if(eleName == "topic")  currentTopic = null;
      }
      public void printTM(Vector topics, int indent){ // not used
            if (topics==null) return;
            indent++;
            //System.out.println("Topic Map-------------------------------------------------
---");
            for (int i=0; i<topics.size(); i++){
                  Topic t = (Topic) topics.elementAt(i);
                  System.out.println(spaces(indent)+t.getID());
                  printTM(t.getSubTypes(), indent);
            }
      }
      // prints just the names of topics in the vector
      public void printTopics(Vector topics){   // not used
            System.out.println("Topics:begin");
            for(int i=0; i<topics.size(); i++)
                  System.out.println(((Topic)topics.elementAt(i)).getID());
            System.out.println("Topics:end");
      }
      public void printTopicMap(Topic t, int indent){
            if (t==null) return;
            System.out.println(spaces(indent) + t.getDisplayName());
            indent += 3;
            if (t.nodetype == 't' || t.nodetype == 'a' || t.nodetype == 'o' || t.nodetype ==
'r') //if a leaf node or a base type
                  return;
            Vector children = t.getSubTypes();
            for (int i=0; i<children.size(); i++)
                  printTopicMap((Topic)children.elementAt(i), indent);
      }
      public String spaces(int indent){
            String temp = "";
            for(int i=0; i<indent; i++)
                  temp += " ";
            return temp;
      }
      public ParseTM(String filename){
            try {
                  XMLReader saxParser = ( XMLReader )
Class.forName("org.apache.xerces.parsers.SAXParser" ).newInstance();
                  saxParser.setContentHandler( this );
                  FileReader reader = new FileReader( filename );
                  saxParser.parse( new InputSource( reader ) );
            }
            catch ( SAXParseException spe ) {System.err.println( "Parse Error: " +
spe.getMessage() );}
            catch ( SAXException se ) {se.printStackTrace();}
            catch ( Exception e ) {e.printStackTrace();}
      }
}
```

TM_Explorer.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Vector;
import javax.swing.tree.DefaultMutableTreeNode;

class TM_Explorer extends JFrame{
      private JPanel    pan1, pan2, pan3, pan4;
      private ParseTM p;      // used for parsing TM and creating a general hierarchy of
everthing in p.topics vector, first element of this vector is the root of the hiearachy
```

```
  public static void main(String[] arg){
      (new TM_Explorer()).setVisible(true);
  }
      public TM_Explorer() {
            // set up menu and 4 panels, menu actionlistener starts the job
            super("Hierarchical Topic Map Explorer");
            this.setSize(800, 650);
            this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            this.getContentPane().setLayout(new BorderLayout());

            //SET UP MAIN MENU-------------------------------------------
            JMenuBar menuBar = new JMenuBar();
            JMenu menu = new JMenu("File");

            menu.setMnemonic(KeyEvent.VK_F);
            menuBar.add(menu);
            //open item
            JMenuItem openItem = new JMenuItem("Open", KeyEvent.VK_O);
            openItem.addActionListener(new OpenMenuItemListener(this)); // Action starts
here
            menu.add(openItem);
            // save menu item
            JMenuItem saveItem = new JMenuItem("Save", KeyEvent.VK_O);
            saveItem.addActionListener(new OpenMenuItemListener(this));
            menu.add(saveItem);
            // quit item
            JMenuItem quitItem = new JMenuItem("Quit", KeyEvent.VK_O);
            quitItem.addActionListener(new OpenMenuItemListener(this));
            menu.add(quitItem);
            this.setJMenuBar(menuBar);
            //SET UP MAIN MENU-------------------------------------------

            // SETUP UP MAIN PANEL IN JFrame
            JPanel mainPanel = new JPanel();
            mainPanel.setLayout(new GridLayout(2, 2, 1, 1));
            mainPanel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));
            ((Container)this.getContentPane()).add(mainPanel, BorderLayout.CENTER);

            // SET UP 4 INDVIDUAL PANES--at the beginning display empty hiearachies
            pan1 = new JPanel();
            pan1.setLayout(new BorderLayout());
        pan1.add(new JTree(new DefaultMutableTreeNode("Topic Type Hierarcy")),
BorderLayout.CENTER);
            mainPanel.add(new JScrollPane(pan1));
        pan2 = new JPanel();
        pan2.setLayout(new BorderLayout());
        pan2.add(new JTree(new DefaultMutableTreeNode("Association Type Hierarcy")),
BorderLayout.CENTER);
        mainPanel.add(new JScrollPane(pan2));
        pan3 = new JPanel();
        pan3.setLayout(new BorderLayout());
        pan3.add(new JTree(new DefaultMutableTreeNode("Role Type Hierarcy")),
BorderLayout.CENTER);
        mainPanel.add(new JScrollPane(pan3));
        pan4 = new JPanel();
//          pan4.add(x);
        pan4.setLayout(new BorderLayout());
        pan4.add(new JTree(new DefaultMutableTreeNode("Occurrence Type Hierarcy")),
BorderLayout.CENTER);
        mainPanel.add(new JScrollPane(pan4));
            repaint();
      }
  public DefaultMutableTreeNode toJTree(Topic t, char type){
      //System.out.println(root.getNodeName());
      DefaultMutableTreeNode r = new DefaultMutableTreeNode(t.getDisplayName());
      if (t.isleafnode()){
            if (t.nodeIsOf(type)){ // is this right type of leaf node
                  //localHierarchies(r, t, type, "local"); // add 3 local hierarchies to
node r
                  return r;
            }else // wrong type of leaf node, stop
                  return null;
      }else{ // internal node
            if (t.nodetypeOK(type)){ // right type of internal node, visit children
                  for (int i = 0; i < t.getSubTypes().size(); i++){
                        DefaultMutableTreeNode temp =
toJTree((Topic)t.getSubTypes().elementAt(i), type);
                        if (temp != null) r.add(temp);
                  }
                  return r;
            }else // wrong type of internal node, stop
                  return null;
      }
  }
  /*
   * creates 3 local hierarchies depending on the type of the Topic t node
   * if t is topic then an assoc, role and occur hierarchies are created as chidren of r
   * if t is assoc then topic, occur, role hierarchies are created as children of r
   */
  public void localHierarchies(DefaultMutableTreeNode r, Topic t, char type, String ename){
      DefaultMutableTreeNode tmp;
      switch (type){
      case 't':
            tmp = localHierarchy(r,t,type,'a',""+ename+" associations");if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'r',""+ename+" roles");       if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'o',""+ename+" occurences");  if (tmp!=null)
r.add(tmp);
            break;
      case 'a':
            tmp = localHierarchy(r,t,type,'t',""+ename+" topics");            if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'r',""+ename+" roles");       if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'o',""+ename+" occurences");  if (tmp!=null)
r.add(tmp);
            break;
      case 'o':
            tmp = localHierarchy(r,t,type,'t',""+ename+" topics");            if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'a',""+ename+" associations");      if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'r',""+ename+" roles");       if (tmp!=null)
r.add(tmp);
            break;
      case 'r':
            tmp = localHierarchy(r,t,type,'t',""+ename+" topics");            if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'a',""+ename+" associations");      if (tmp!=null)
r.add(tmp);
            tmp = localHierarchy(r,t,type,'o',""+ename+" occurences");  if (tmp!=null)
r.add(tmp);
      }
  }
  public DefaultMutableTreeNode localHierarchy(DefaultMutableTreeNode r, Topic t, char
globalType, char localType, String ename){
      /*
       * creates a local hierarchy of 'localType' for the t node in global hierarcy of
'globaltype'
       */
      Vector nodelist = new Vector();
      nodelist = getNodesOfType (t, globalType, localType); // get a list of leaf nodes
      CreateHiearchyByTraversingBottomUp(nodelist); //at this point the hierarchy is stored
in p.topics vector with flag field is set true
      p.tm.setId(ename);
      return convertHierarcyToJtree(p.tm); // traverse top down from the root, create a
JTree version of hierarchy
  }
  public void printXNodeList(DefaultMutableTreeNode r, Vector v, Topic t){
      for(int i=0; i<v.size(); i++)
            r.add(new DefaultMutableTreeNode( ((Topic)v.elementAt(i)).getDisplayName() ));
  }
  public DefaultMutableTreeNode convertHierarcyToJtree (Topic t){
      if (!t.flag)      return null; // this line is not necessary
      DefaultMutableTreeNode node = new DefaultMutableTreeNode(t.getID());
      if (t.isleafnode())
            return node;
      Vector children = t.getSubTypes();
      if (children==null)
            return node; // this is not suppoed to be, becuse it must be otherwise be a leaf
node
      for (int i=0; i<children.size(); i++){
            Topic c = (Topic)children.elementAt(i);
            if (c.flag)
                  node.add(convertHierarcyToJtree(c));
      }
      return node;
  }
  public Vector getNodesOfType (Topic t, char globalType, char localType){
  /*
   * scannes the topics vector for nodes() of 'type'  incident upon topic t and
   * returns a vector of such nodes, when creating a local hierarchy of type
   *
   */
  if (t.isleafnode())
      switch(globalType){
      case 't':   switch(localType){
                        case 'o': return t.getSubTypes();
                        case 'a': return getAssociationsOfTopic (t);
                        case 'r': return getRolesOfTopic (t);
                        }
                        break;
      case 'a':   switch(localType){
                        case 't': return t.getSubTypes();
                        case 'r': return ((Association)t).getRoles();
                        case 'o': return combineLists(t.getSubTypes());
                        }
                        break;
      case 'r':   switch(localType){
                        case 'a': return t.getSubTypes();
                        case 't': return combineLists(t.getSubTypes());
                        case 'o': return combineLists(combineLists(t.getSubTypes()));
                        }
                        break;
      case 'o':   switch(localType){
                        case 't': return t.getSubTypes();
                        case 'a': return getAssociationsOfTopics (t.getSubTypes());
                        case 'r': return getRolesOfTopics        (t.getSubTypes());
                        }
                        break;
      }
  else {// t is not leaf/base/instance, t is inner/type node
      Vector tmp = new Vector();
      Vector children = t.getSubTypes();
      if (children==null){
            System.out.println("Error in getNodesofType, inner node must have children");
            return null;
      }
      for(int i=0; i<children.size(); i++){
            Topic aux = (Topic)children.elementAt(i);
            if ((aux.isleafnode() && aux.nodetype==globalType) ||
                (!aux.isleafnode()) && aux.nodetypeOK(globalType))
                  tmp.addAll(getNodesOfType(((Topic)children.elementAt(i)), globalType,
localType));
      }
      return tmp;
  }
      System.out.println("Error in GetNodesOfType!");
      return null;
  }
  public Vector getAssociationsOfTopics (Vector topicList){
      Vector tmp = new Vector();
      for (int i = 0; i < topicList.size(); i++) // no need to check for the correctness of
nodetype of nodes in topicList
            tmp.addAll(getAssociationsOfTopic((Topic)topicList.elementAt(i)));
      return tmp;
  }
  public Vector getRolesOfTopics (Vector topicList){
      Vector tmp = new Vector();
      for (int i = 0; i < topicList.size(); i++)
            tmp.addAll(getRolesOfTopic((Topic)topicList.elementAt(i)));
      return tmp;
  }
  public Vector getAssociationsOfTopic(Topic t){
      /*
       * scan the topic map to return a list of association containing topic t
       */
      Vector List = new Vector();
      for (int i = 0; i < p.topics.size(); i++)
            if (((Topic)p.topics.elementAt(i)).nodetype == 'a'){
                  Topic a = (Topic)p.topics.elementAt(i);
                  if (t.containedin(a.getSubTypes()) != -1)
                        List.add(a);
            }
      return List;
  }
  public Vector getRolesOfTopic(Topic t){
      /*
       * scan the topic map to return a list of roles containing topic t
       */
      Vector List = new Vector();
      for (int i = 0; i < p.topics.size(); i++)
            if (((Topic)p.topics.elementAt(i)).nodetype == 'r'){
                  Topic a = (Topic)p.topics.elementAt(i);
                  if (t.containedin(combineLists(a.getSubTypes())) != -1)
                        List.add(a);
            }
      return List;
  }
  public Vector combineLists(Vector v){
      /*
       * combines the subtypes of the objects in vector v into a new vector
       */
      Vector tmp = new Vector();
      for (int i = 0; i < v.size(); i++)
            tmp.addAll(((Topic)v.elementAt(i)).getSubTypes());
      return tmp;
  }
  public void CreateHiearchyByTraversingBottomUp (Vector list){
      clearFlags(p.topics); // clear flag before stting them
      if (list==null) return;
      for(int i = 0; i < list.size(); i++)
            setFlagBottomUp((Topic)list.elementAt(i));
  }
  public void clearFlags(Vector topics){
      for(int i=0; i<topics.size(); i++)
            ((Topic)topics.elementAt(i)).flag = false;
  }
  public void setFlagBottomUp(Topic t){
      if (t.flag) return;     // if already set,
      t.flag = true;
      Vector parents = t.getSuperTypes();
      if (parents==null)      return;
      for(int i = 0; i < parents.size(); i++)
            setFlagBottomUp((Topic)parents.elementAt(i));
  }

  public DefaultMutableTreeNode processAllNodes(Topic t){
      //Not used, displays Topic Map Type Hierarchy
      DefaultMutableTreeNode r = new DefaultMutableTreeNode(t.getDisplayName());
      if (t.isleafnode()){
                  return r;   //create and return a node for t
      }else{ // internal node
                  for (int i = 0; i < t.getSubTypes().size(); i++){
                        DefaultMutableTreeNode temp = 
processAllNodes((Topic)t.getSubTypes().elementAt(i));
                        if (temp != null) r.add(temp);
                  }
                  return r;
      }
  }
  public void addStatisticsToTree(DefaultMutableTreeNode r){
      /*
       * adds following parameters to the name of r node
       * r is the root of a tree
       * Parameters:
       *    depth of subtree of r
       *    no of nodes in subtree
       */
    }
  public class ActionListenerYYY implements ActionListener {
      public ActionListenerYYY(JFrame p) {
            //this.parent = p;
    }
    public void actionPerformed(ActionEvent e) {
      System.out.println("cc");
    }
  }
  public class OpenMenuItemListener implements ActionListener {
      private JFrame parent;
      public OpenMenuItemListener(JFrame parent) {
            this.parent = parent;
    }
    public void actionPerformed(ActionEvent e) {
      JFileChooser chooser = new JFileChooser();
      int returnVal = chooser.showOpenDialog(parent);
      if (returnVal == JFileChooser.APPROVE_OPTION) {
            p = new ParseTM(chooser.getSelectedFile().getName());
            //now p.topics.firstElement() contains the root of the Topic Map
            DefaultMutableTreeNode  tth, ath, rth, oth;

            // create tth, ath, rth and oth
            ((Topic)p.topics.firstElement()).setId("Topics");
            tth = toJTree( (Topic)p.topics.firstElement(), 't');
            addStatisticsToTree(tth);

            ((Topic)p.topics.firstElement()).setId("Associations");
            ath = toJTree( (Topic)p.topics.firstElement(), 'a');
            ((Topic)p.topics.firstElement()).setId("Roles");
            rth = toJTree( (Topic)p.topics.firstElement(), 'r');
            ((Topic)p.topics.firstElement()).setId("Occurrences");
            oth = toJTree( (Topic)p.topics.firstElement(), 'o');
                  // UPDATE GUI
            pan1.removeAll();
              pan1.add(new JTree(tth), BorderLayout.CENTER);
              pan1.validate();

            pan2.removeAll();
              pan2.add(new JTree(ath), BorderLayout.CENTER);
              pan2.validate();

            pan3.removeAll();
              pan3.add(new JTree(rth), BorderLayout.CENTER);
              pan3.validate();

            pan4.removeAll();
              pan4.add(new JTree(oth), BorderLayout.CENTER);
              pan4.validate();
                  repaint();
      }
    }
 }
}
```

APPENDIX D

ABBREVIATIONS

SYMBOL / ABBREVIATION

XML	Extensible Markup Language
XTM	XML Topic Maps
TM	Topic Map
TTH	Topic Type Hierarchy
ATH	Association Type Hierarchy
RTH	Role Type Hierarchy
OTH	Occurrences Type Hierarchy
DTD	Document Type Definitions
TT	Topic Type
AT	Association Type
RT	Role Type
OT	Occurrences Type
TO	Topic Object
GUI	Graphical User Interface