Enabling knowledge management of organizational memory for groups through shared topic maps

by

Alok Sharma

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Leslie Miller, Major Professor
Shashi K. Gadia
Sree Nilakanta

Iowa State University

Ames, Iowa

2010

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

An Organization's success and growth are increasingly becoming dependent upon how efficiently the Business Processes are managed. Knowledge Management and Organizational Memory have become critical components in the management of Business Processes. As, the Business Process Managers, Participants and End Users are dependent upon information and knowledge from myriad sources, thus their ability to proficiently access, use and generate new knowledge from existing knowledge has become quite vital. Organizations today work as an interconnected web of their Business Processes. Thus to ensure success, to garner the enhanced organizational capabilities and to have sustain performance improvements, they should incorporate knowledge-sharing among different teams. We present a model, based on Shared Group Topic Map approach which uses these concepts and illustrate its use by way of supporting a group search implementation. The design, implementation and example scenario of this system are given and explained.

CHAPTER 1. OVERVIEW

1.1 Introduction

In today's competitive environment, organizations are operating as an interconnected web of their Business Processes. These Business Processes span across departmental and organizational boundaries. Thus Business Processes of an organization are becoming more and more interconnected. In such an environment the role of the employees in performing these processes efficiently is becoming increasingly diffuse. Thus, to ensure success with these business processes, firms should proficiently incorporate knowledge-sharing among managers of the hubs as a precursor to integration and unification of these processes.

A key concept in the efficient management of Business Processes is that the actionable knowledge is created through distributed cognition and shared practices. This knowledge and information is generated from a variety of sources and channels. Moreover, it is collected and maintained as an ever expanding repository of Organizational Memory. Accessing and using this vast Organizational Memory, is a challenge because of the multifaceted nature of memories and knowledge sources. Another critical challenge is how to incorporate the ever growing knowledge in a distributed cognition environment while maintaining consistency, durability and correctness.

To overcome these challenges we present a model based on the topic maps. In this approach pertinent information is categorized into topics that are associated with the processes which key actors find valuable. The topic map based model facilitates organizing, navigating and searching the knowledge and information in incessantly growing and distributed organizational memories. Moreover to facilitate distributed cognition, shared group topic maps are provided. These are used throughout the organization across different teams. Multiple users belonging to different teams can use these shared group topic maps simultaneously. A team member may have access to multiple topic maps. All these enables team members in performing cross functional and diffuse roles. Moreover to cater the ever increasing knowledge in the distributed environment, we have provided dynamic structure to the shared group topic maps. Different team members can modify the group topic maps to reflect the latest state of their cognition. They can perform add, create, move, change and delete operations on the various information sources. Moreover, tools like information retrieval tool, frequency analysis etc are provided for supporting a group search implementation.

1.2 Overview of the Report

The next Chapter presents a brief review of the relevant literature. An overview of the system model is covered in Chapter 3. Chapter 4 provides implementation detail and an example scenario. Conclusions and future work are addressed in Chapter 5.

CHAPTER 2. BACKGROUND

The Organizational Memory topic encompasses the stored information of an organization. It includes all the stored records, all the institution created internal documentation covering patents, copyrights, trademarks, brands, registered design, trade secrets and processes. It also includes knowledge and the expertise of current employees, their task environments, artifacts and the tools required to locate and/or interpret available information [1, 2, 3, 4, 5, 6]. It can be used for information retention, acquisition, and retrieval in an organization [7].

As Organizational Memory can be a large and valuable repository of information and knowledge. It can play a pivotal role in effecting organizational performance [8, 9, 10, 11, 12]. The organizational memory may help the managers to maintain strategic plans over time. It enables referring to old solutions for solving new problems. It can facilitate the access of organizational expertise to naïve workers [13]. It is necessary for organizational learning and adaptation [14]. Moreover, organizational know-how can be seen as a new source of competitive advantage [15].

Organizational Memory is distributed across different conveniences it is not stored centrally. Organizational memory is necessarily distributed through time and across not only group members, but groups. Organizations operate as an interconnected web of their business processes. To ensure success with business processes, firms need to more efficiently incorporate knowledge-sharing among managers of the hubs as a precursor to integration of processes. Only then can firms reap the enhanced organizational capabilities and sustain performance improvements in these processes [16]. Thus Ackerman and Halverson [17] used the theory of distributed cognition to develop a theoretical foundation for organizational memory. The theory of distributed cognition was developed by Hutchins and Hollan [18,19]. Distributed cognition refers to the perception that cognition occurs in a distributed manner. Distributed cognition theory is concerned with the organization and operation of cognitive systems. It deals with the mechanisms that make up cognitive processes, which results in cognitive accomplishments. Furthermore, it re-situates cognition in the socio-cultural context. Ackerman and Halverson [17] claim that as socio-technical systems, organizations and their memories conform to social structures and norms while employing technical models. The basic tenets of this theory are that knowledge evolves from a community of practice and that cognition and inferences result from the shared meaning among the participants (hence the distribution) [18, 19]. Communities of practice fulfill a number of functions with respect to the creation, accumulation, and diffusion of knowledge in an organization through exchange and interpretation of information, by

retaining knowledge, by stewarding competencies, and providing homes for identities [20]. Collective thinking creates knowledge that otherwise would not be evident. Using empirical data and qualitative methods, Ackerman and Halverson [17] illustrate the application of the theory of distributed cognition to validate the use of organizational memory in decision making.

While addressing distributed cognition in organizational memory several challenges have to be overcome. Like, ever growing organizational memories, lack of powerful and effective mechanisms and functionalities for navigating, linking, searching and investigating, the need for enhanced access mechanisms. To overcome these, the ISO standard ISO/IEC 13250 Topic Maps [22] defines a model and architecture for the semantic structuring of link networks. Topic maps help in finding relevant knowledge or information in continuously growing and distributed organizational memories. Smolnik and Erdman [21] state that as topic maps provide strong paradigms and concepts for the semantic structuring of link networks therefore, they are a considerable solution for organizing and navigating large and, continuously growing network of organizational memories.

2.1. Topic Maps

Miller et all[5] claim that as social and linguistic dimensions of knowledge become pertinent, users need appropriate tools to support the search and analysis of knowledge. Topic maps are one such tool. Topic maps address the collaborative aspect of process management and group support. By applying Topic maps on large set of heterogeneous information resources, reusable structured semantic link network are created. The topic maps enable users to search knowledge structures and selectively navigate to the required knowledge objects, independent of their origin [23].

Topic maps provide a subject based classification of resources where each resource represents a real world object or tool. In topic maps, three constructs are provided for describing the subjects represented by the topics: names, occurrences, and associations. These describe the names, properties, and relationships of subjects, respectively. A name may be assigned to more than one topic and a topic may have more than one name. Also by defining scope and types a name can become rich and complex. An occurrence links to one or more real knowledge sources. An occurrence, however, is not part of the topic map. Associations describe relationships among topics and are independent of the real knowledge objects. Associations add value through the relationships. The ISO standard ISO/IEC 13250 [22] 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. The standard further states that 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.

The Topic Map architecture is designed to help merging topic maps without needing the merged topic maps to be copied or modified. This feature makes it suitable for use in organizational memory systems. As Steiner et al [24] state, constructing large, complex organizational memories takes place over extended periods of time, growing incrementally both in size and structure. The ISO standard[22] describes how topic maps may be used in situations such as that may be found in OM systems. For example, a topic map can be employed "to structure unstructured information objects, or to help create topic-oriented user interfaces that provide the effect of merging unstructured information bases with structured ones [25]. The overlay mechanism of topic maps can be considered as an external markup mechanism, in that an arbitrary structure is imposed on the information without changing its original form." Knowledge management applications can also benefit from topic maps, especially when combined with ontologies [26]. Korthaus et al. [27] describe a topic grid architecture that enables a client application to view distributed topic maps of organizational knowledge as a combined virtual topic map. Each distributed topic map represents a different view of the underlying information.

Miller et. al [5] developed a model of a topic map based OM knowledge management system to represent and use organizational memory artifacts. Their paper presented the design and development of an organizational memory warehouse that supports the distributed cognition theory.

We ameliorate and expand on the model presented by Miller et al [5]. In their model, the team members had access to just a single topic map with no provisions for incorporating topic maps from other users. As a result knowledge sharing was minimal. Moreover, their model didn't allow knowledge growth as only Process Topic Maps were used. The Process Topic Maps are static and no modifications are allowed because they have been designed to provide support for individuals working on processes that are defined on organization wide basis. Our model of topic map implementation addresses the collaborative aspect of process management and group support. In our model, a team member may incorporate and use Topic maps from other users. Thus it enables knowledge sharing. Our model also supports knowledge growth as it supports both the Process Topic

Maps and Group Topic Maps. Group Topic Maps enables team members to perform various modification operations. Thus, the user can modify the topic map structures to capture and reflect his/her current cognition by adding, creating, duplicating, modifying, moving and deleting different knowledge sources. Moreover tools like information retrieval tool, frequency analysis etc are provided for supporting a group search implementation.

In this chapter we provided brief overview of the background and the theoretical concepts involved. We also briefly compared the relevant past models with our approach. In the next chapter we present our system model.

CHAPTER 3. SYSTEM MODEL

Every organization tries to manage the business knowledge that it has accumulated over the years in the best possible manner. To achieve this, it is necessary to determine the sources of business knowledge and provide members of the organization access to all sources in a timely matter. In our model we have focused on the latter issue of providing efficient access through retrieval, analysis, merging, modifying, and learning tools. We have assumed that business knowledge sources are employee knowledge, transaction data, data warehouse data, memory artifacts (internal and external), and the knowledge inferred from these sources. In our system we have used Topic Maps to achieve these goals.

3.1Topic Maps

A topic map is used to provide users with visual to both search and analysis. It is this interesting blend of semantic search and analysis that motivates its use in the proposed model.

Here we define a topic map to be the directed acyclic graph T = (N,E), where N is the set of topic map terms, data types, or search/analysis tools. The directed edges in E are of the type (n1,n2), where n1 is a topic map term or a data type and n2 is a topic map term, a data type, or a search/analysis tool. An implicit node, called the root, points to the topic map terms, data types and/or tools that make up the first level of the topic map. Nodes without degree zero are said to be leaves of the topic map. Leaves point to the information in the organization's memory and carry any search terms and/or data types accumulated on the path from the root of the topic map to the leaf. Note that leaves that are either search terms or data types make use of an implicit search tool or a list of resources. In addition each non-leaf level of the topic map has a search tool that can be used to initiate a search based on the topic map terms that have been traversed to get to the current topic map level. A visual topic map over a topic map T = (N,E) can be seen as a directed acyclic graph V = (S,A), where S is a set of screens and A is a set of directed edges that connect the screens via a node context. A screen s ε S is defined as 2-dimensional representation of a set of nodes s_N such that each node m in s_N is in N and there is a node in n ε N such that (n,m) ε E for every m in s_N . An edge exists in A whenever a non-leaf topic node on one screen points (in T) to the nodes on the second screen. The root node described in the topic map definition above is replaced by the screen representing the nodes in T that are adjacent to the implicit root node. Each edge in A has a node context. This node context

implies selecting a non-leaf node on a screen results in following the link to the screen associated with the node. In our system we model a Topic Map by storing the details of the set of screens present in that Topic Map and by storing the details of the nodes present in a particular screen. Figure 3 illustrates a screen shot of a screen in the current prototype.

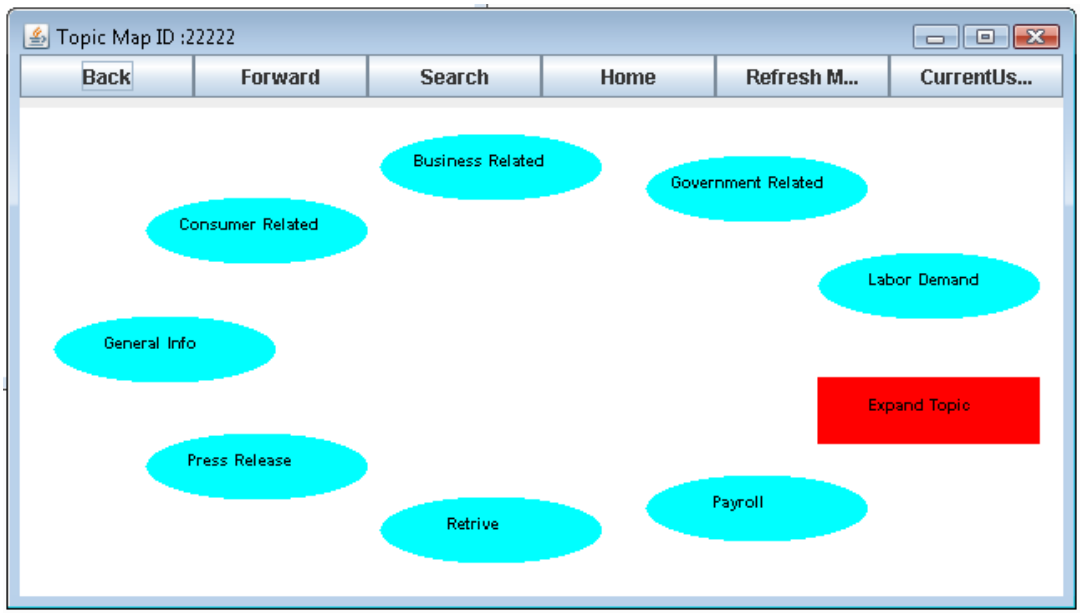

Figure 3.1 Sample Screen in the current prototype.

3.2 Process and Group Topic Maps

The proposed model makes use of two types of topic maps, namely a Process Topic Map and a Group Topic Map, as a means of managing user's access to the knowledge available to the organization through its organizational memory.

3.2.1 Process Topic Map

The Process Topic Map is defined by the pair $< V, p>$, where V is a visual topic map and p indicates the business process on which the topic map has been constructed. We use the Visual Topic Map Model we discussed in the section 3.1 to model the Process Topic Map.

The path label of node in a topic map indicates the union of the topic terms from the root to the node. Actors and roles are associated with each process. The roles include process owner, process manager, and process participants, where each one has different knowledge requirements. From a user's point of view the structure of the Process Topic Map is fixed. The motivation for using such a static view of the Process Topic Maps is that they have been designed to provide support for

individuals working on processes that are defined on an organization wide basis. The one deviation from this approach is based on the level of users. When building a topic map, one has to make a decision concerning who is the target audience. The main question is how to deal with the experience range of the users. For a topic map to be useful to a novice user, it is necessary that it has a sufficiently fine granulation so that the novice user will be able to make use of the topic map. The result is that experienced users will have to waste time moving down through the screens in order to get to the topics and/or tools that they need to use. To bridge this gap, we have introduced the notions of a shortcut node and a corresponding shortcut screen. As a user gains experience in working with a process and the related topic map, he/she can make shallow copies of topics and/or tools and place them on the shortcut screen. For convenience, we have placed the shortcut node on the first user screen. The user simply has to indicate the desire to create a shortcut for the node. In our current prototype, the user right clicks on the node to make the copy. The node's path label and screen link (for non-leaf nodes) are copied to the shortcut screen along with the node. As a result, the experienced user can have access to any topic and/or tool without having to traverse the topic map. As the needs of the user changes they can delete any nodes that they no longer need access to. The model supports one individualized shortcut screen for each Process Topic Map that a user has access to. Although a novice user may navigate the map from its "root", an expert user might want to use a "shortcut" screen. The shortcut screen provides quick access to the part of the topic map structure that the user often uses. The shortcut screen would include nodes where each node might be a path to further levels of the topic map or calls to various tools and services. The process topic map and the user's (actor/role) view of the topic map persist in the system.

3.2.2 Group Topic Map

While Process Topic Maps are useful for individuals working on the processes that they have been assigned, group tasks (e.g., committees) that bring together people from across the organization require an approach to managing knowledge for the group. To deal with these concerns, our model introduces the concept of a Group Topic Map. A *Group Topic Map* is defined by the 4-tuple < *V,G,O,R*>, where *V* is a visual topic map, *G* is the group of users, *O* is the set of operations defined on *V,* and *R* is the set of roles relevant to the group's mission.

The visual topic map in a Group Topic Map has properties similar to the visual topic map in the Process Topic Map. The basic definition of a visual topic map remains unchanged. It incorporates the shortcut node and screen for the same reasons given above. The primary difference is that the notion of a path label is weakened in this version of the visual topic map. The motivation for this

difference is that it gives the model the opportunity of making use of the users' understanding of their areas of work and skills. It gives them the ability to place nodes on screens that they feel will provide the best information for other members of the group. To this point, we have assumed that everyone in the group can use all of the operations in *O*, but our prototype could easily be extended to restrict who can make the changes and limit the kinds of operations that individual users can perform.

The initial visual topic map would be up to the group. It could simply be an empty initial screen, a topic map that exists based on prior efforts of the group, or a topic map that is created by members of the group using the operations available in the operation set. The choice would be up to the group and/or the organization at large. In the remainder of this section, we describe the different components of the model.

3.3 Group

The group represents the set of members of the group. In our current model this is simply a list of the people that can use and manipulate the Group Topic Map. It could easily be extended to include information on privileges assigned to the individual members of the group. In our model, the administrator assigns different Topic Maps , different roles to the users.

3.4 Operations

The operations (*O*) defined for the Group Topic Map operate on the nodes of the topic map. The focus of the operations is to allow the individual members of the group to impact the group's access to the organization's knowledge by modifying the group's search process through changes in the Group Topic Map set up to support the group's use of organizational knowledge. The basic operations available to members of the group are:

AddANode – The operation allows users to copy nodes from the Process Topic Maps to the Group Topic Map. This operation is based on the processes that the user has access to for their non-group activities and group activities. The system enables the user to right click on the node he wants to add and select the AddANode operation. Then, the user selects the specific Group Topic Map and navigates to the screen he wants to add this node, on the desired screen the user right click on the empty space and selects the copy node operation. The system copies node and its descendents (if any) to the screen along with their current path labels and screen linkage. This is enabled by invoking the copy operation.

In the copy operation, the system recursively combines the copied node's details and it's next screen linkages details with the topic map id and the screen locator of the destination Group Topic map and Screen and stores this information in the database. This process continues recursively till the leaf nodes are processed.

CreateANode – This operation is used to add new knowledge to the Group Topic Map. Whereas, the AddANode operation copies an existing node and its associated knowledge into the Group Topic Map, the CreateANode is used to add either newly created or synthesized knowledge. It is well-known that there is always a difference between the true organizational memory knowledge (TOM) and the accessible organizational memory knowledge (AOM). The gap, TOM - AOM includes knowledge that is hard to capture as well as knowledge that is sufficiently new that it has yet to be captured. The CreateANode operation allows a user to capture some of this knowledge and provide access to it through the Group Topic Map. In our system, the user can create either a new Topic or Tool. The system enables the user to select the topic map and its screen where he wants to create a new node. The user right clicks on the screen he wants to create a node and selects the CreateANode operation. Then the system provides the user with the option to create either a Topic or Tool. In the current model, the system enables the user to provide the screen term for the new node. If the new node is a Topic, then the system provides the user with the list of resources. The user selects the resource he wants to associate with the new node. Similarly, in case of a Tool, the system provides the user with the list of available tool and the user selects a tool to be associated with the node.

GeneralizeANode – This operation is designed to allow users to partition the knowledge space indicated by the semantics of a node. It is used when the user sees the need for a finer level of detail in dividing the knowledge space.

The GeneralizeANode operation creates a new node. The new node reflects the underlying concepts of the original node while providing a more generalized cognition. The system can use the existing knowledge indicated by the original node combined with the definition of the new node to search the knowledge space. The results of the search would reflect knowledge which is more generic. The GeneralizeANode operation helps to provide a broader interpretation of the copied node’s knowledge.

For example, in an organization that develops software; different participants may be present like a project manager, software programmer, software tester, etc. Each of these performs specific individual processes and may be participating in organization wide group process. In our system,

corresponding to the individual processes we would have process topic maps and corresponding to the group processes we would have the shared group topic maps. Now, a software programmer may copy a node from his individual process topic map to the shared group topic map. Thus, all the information and knowledge specific to the software programming process represented by the acyclic graph beneath this node would be incorporated in the shared group topic map. The knowledge represented by this node and its descendants would be more skewed to the software programming process. By using the GeneralizeANode operation we can create a new node which is more generalized and which may be more useful to other process participants like project manager or software tester which may want to use the standard information.

The user performs the GeneralizeANode operation by right clicking on the node. The system provides the user with the text field to enter additional search terms. The user enters the search terms separated by commas. When the user clicks on the submit button, a new node is created on the same screen as the node on which the GeneralizeANode operation was performed. The screen name of the new node is created by concatenating the keyword "Generalized" and the screen term of the node on which the GeneralizeANode operation was performed. The information retrieval tool is associated with the newly created node. The path labels to the current screen and the additional search terms entered by the user are used by the information retrieval tool when the user clicks on the newly created node. In our current implementation, the GeneralizeANode operation can't be performed on the leaf nodes i.e Tool nodes or Topic nodes that have resources associated with them.

SpecifyANode(Role) – A user may copy a node from the Process Topic Map to Group Topic Map or from a Group Topic Map to another Group Topic Map for his group related cognition. The newly copied node to some extent is associated with knowledge of the underlying process of the Topic Map from which it was copied. In our system, a user has a specific role assigned for a particular topic map. Corresponding to this role some metadata is associated. The SpecifyANode operation creates a new node. The new node reflects the underlying concepts of the original copied node while incorporating the knowledge and cognition related to the role of the user. The system can use the existing knowledge indicated by the original node combined with the definition of the new node to search the knowledge space. The results of the search would reflect knowledge more skewed to the new role.

For example, if we take the same organization as we discussed in the GeneralizeANode operation, then a software programmer may copy a node from a group or a process topic map he/she has access to a shared group topic map. Thus, all the information and knowledge specific to the

original process and represented by the acyclic graph beneath this node would be incorporated in the shared group topic map. In SpecifyANode operation, the metadata associated with the user role is used to create a new node which is more specific and which may be more useful to software programmer role as compared to the standard information.

The user performs the SpecifyANode operation by right clicking on the node. The system provides the user with the text field to enter additional search terms. The user enters the search terms separated by commas. When the user clicks on the submit button, a new node is created on the same screen as the node on which the SpecifyANode operation was performed. The screen name of the new node is created by concatenating the keyword "Specialized" and the screen term of the node on which the SpecifyANode operation was performed. The information retrieval tool is associated with the newly created node. The path labels to the current screen, the metadata associated with the user role and the additional search terms entered by the user are used by the information retrieval tool when the user clicks on the newly created node. In our current implementation, the SpecifyANode operation can't be performed on the leaf nodes i.e Tool nodes or Topic nodes that have resources associated with them.

ModifyANode – This operation allows the user to modify attributes of a node. The user right clicks on the node he wants modify and from the popup menu, selects ModifyANode operation. After that the system responds by giving the user the choice to change the screen term, change the path label, change the tool and change the resource list. When the user changes the screen term, the system provides the user a text field to provide new screen term, the new screen term is used by the system to determine the path label from the root when the search tool is used for the current node or any of its descendants. When the user changes the path label, the system provides the user a text field with the old path label, the user can modify it; the new path label is used by the system when the search tool is used for the current node or any of its descendants. The system allows changing the tool and the resource list to be used only with leaf nodes. When the user changes the tool, the system provides the user with a list to choose the new tool, the newly selected tool is now associated with this node. When the user changes the resource associated with the node, the system provides the user with the list of resources, the user selects the resource he wants to associate with the node.

MoveANode – The motivation is to allow users to move nodes to another screen to enhance the use of the node in the group search process. The user right clicks on the node he wants to move and from the popup menu selects the MoveANode operation. The system then allows the user to select the

screen on which he wants to move this new node. The node from the earlier screen is removed and moved to the new screen along with all the descendants' linkages and nodes. The operation is enables by invoking copy node operation which copies the nodes recursively, until the leaf nodes are processed. When all the nodes are copied recursively then the nodes from the original screen are removed recursively.

DuplicateANode – This operation allows a node to be duplicated on another screen. The user right click on the node he wants to duplicate and from the pop menu selects the DuplicateANode operation. The system then allows the user to select the screen on which he wants to copy this new node. The node is copied to the new screen along with all the descendants' linkages and nodes. The new node is only a shallow copy of the original node. As a shallow copy is made the changes made to the copied nodes and its descendants would not be reflected in the original node and its descendants and vice versa. This process is enabled by recursive calling of the copy node function, which continues till the leaf node is processed.

DeleteANode – This operation enables the users to delete extraneous nodes. The user right click on the node he wants to delete and from the popup menu selects the DeleteANode operation. The system removes the node locally, if same node was present in other Topic Maps the node would be retained in those. Every TopicMap's node details are stored separately. The node, its descendants and screen linkage details are removed from the database recursively until the leaf node is processed.

3.5 Roles

The roles in R are the roles of the members of the group that are relevant to the task that the group is charged with. The set restricts the way some of the operations (e.g., SpecifyANode) work and also add information to potential searches required for operation on individual nodes. In our system corresponding to every role we have certain keywords associated. In operations such as GeneralizeANode the keywords corresponding to the role the user has are used.

In this chapter we provided brief overview of the model and discussed in-depth the important components. In the next chapter we would discuss the Implementation and an Example Scenario.

CHAPTER 4. IMPLEMENTATION

In Chapters 2 and Chapter 3, we presented overview of our work and provided details of our system's model. This chapter describes the implementation of our system and how a user can use it to perform operations that would help to incorporate distributed cognition.

We would discuss the Hibernate based Internal Object Oriented Database which persist the user and topic map details, the details of the package structure and the java classes used in our implementation, AXIS 2 based client server communication, Swing based user interface, the implementation and working of group topic map operations, web based administrative tool and a comprehensive example scenario in this chapter.

4.1 Internal Object Oriented Database

The Internal Object Oriented Database stores the Users, Topic Maps and Data Sources details. It stores the username and password of different users, details of topic maps a particular user has access to, roles and the keywords associated with these roles, list of available tools and resources, versioning details of topic maps and current structural details of the group and process topic map.

The Internal Object Oriented Database is implemented using the Relational Database Management System - MySQL 3.2 and Hibernate. "Hibernate is a powerful, high performance object/relational persistence and query service. Hibernate lets you develop persistent classes following object-oriented idiom - including association, inheritance, polymorphism, composition, and collections " [28]. Hibernate solves Object-Relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling methods. It relieves the developer from manual result set handling and object conversion. It keeps the application portable to all supported SQL databases, with database portability delivered at very little performance overhead. Simple text based Configuration files can be modified for migration from one database to the other, unlike JDBC where java code has to modified and recompiled. Hibernate also provides data query and retrieval facilities through the Hibernate Query Language. A high-level view of the Hibernate architecture is shown in Figure 4.1.

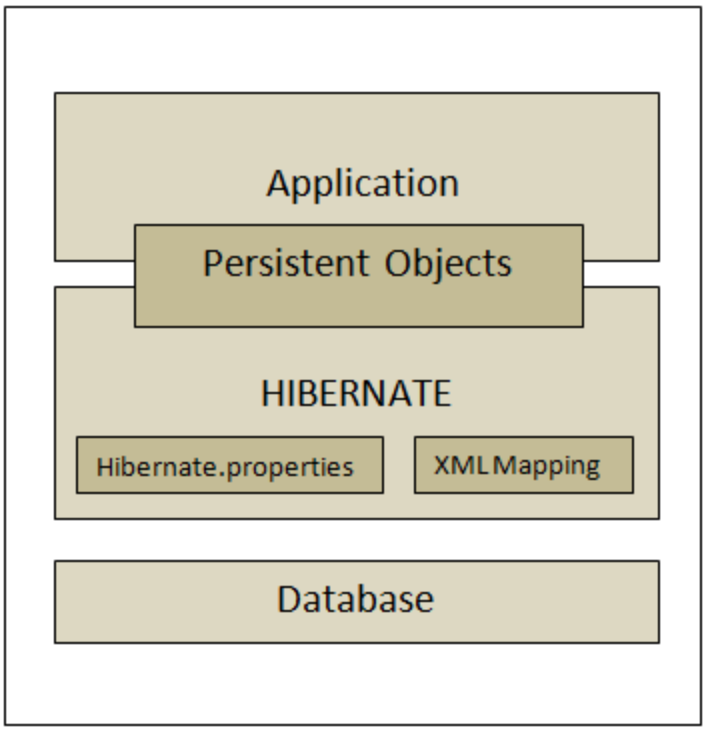

Figure 4.1 Hibernate Architecture [28].

From the figure 4.1, we can observe that Hibernate acts as mediator in the persistence of POJO in Relational Database. Figure 4.1 shows a sample configuration file used in our system. Hibernate make use of configuration and mapping files for providing persistence.

The Hibernate configuration file should have the following properties:

- connection.driver_class -The JDBC connection class for the specific database
- connection.url -The full JDBC URL to the database
- connection.username -The username used to connect to the database
- connection.password -The password used to authenticate the username
- dialect -The name of the SQL dialect for the database
- mapping – The name of hibernate mapping file , which maps POJOs to Tables.

The Figure 4.2 shows a sample Hibernate configuration file used in our implementation.

```
<?xml version='1.0' encoding='utf-8'?>
   <!DOCTYPE hibernate-configuration PUBLI "-//Hibernate/Hibernate Configuration DTD
3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
        <session-factory>
         <property name="connection.url">
                        jdbc:mysql://localhost:3306/topicmap
         </property>
         <property name="connection.driver_class">
                        com.mysql.jdbc.Driver
         </property>
         <property name="dialect">
                        org.hibernate.dialect.MySQLDialect
         </property>
         <property name="connection.username">
                        root
         </property>
         <property name="connection.password">
                        nitish@28NOV
         </property>
                <mapping resource="dataTypes/Screen.hbm.xml" />
                <mapping resource="dataTypes/Node.hbm.xml" />
                <mapping resource="dataTypes/EditTopicMap.hbm.xml" />
                <mapping resource="dataTypes/CurrentUsers.hbm.xml" />
                <mapping resource="dataTypes/GenerateNodeNo.hbm.xml" />
                <mapping resource="dataTypes/ToolTopicList.hbm.xml" />
                <mapping resource="dataTypes/Role.hbm.xml" />
                <mapping resource="dataTypes/UserDetail.hbm.xml" />
        </session-factory>
</hibernate-configuration>
```

Figure 4.2 Example of a Hibernate configuration file.

A Hibernate mapping file provides the mapping between a Java POJO class and a relational database table. Where each attributes of the class is mapped to a column of the corresponding table. Mapping files are used to provide Hibernate with information to persist objects to a relational database. The mapping files are also used to create the database schema automatically. The naming convention for mapping files is to use the name of the persistent class with the hbm.xml extension. POJOs should have getter and setter methods for every attribute in the class and a no argument constructor. To load, store and retrieve Java objects from the internal object oriented database, our system has Java bean classes like *Screen, Node, Role, UserDetail* etc, along with their corresponding XML mapping files. Corresponding to each of these classes, tables are created in the *internal object oriented database.* Figure 4.3 and Figure 4.4 show the Java class *Role.java* and the corresponding XML mapping file *Role.hbm.xml* used in our application.

```
package dataTypes;
public class Role {

        private String roleName ;
        private String keyWords;

        public Role() { }
        public String getRoleName() {
                return roleName;
        }
        public void setRoleName(String roleName) {
                this.roleName = roleName;
        }
        public String getKeyWords() {
                return keyWords;
        }
        public void setKeyWords(String keyWords) {
                this.keyWords =keyWords;
        }

}
```

Figure 4.3 Role.java class.

```
<?xml version="1.0"?>
 <!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd" >

<hibernate-mapping>

        <class name="DataTypes.Role" table="Role" discriminator-value="PARENT">

                <id name="roleName" type="java.lang.String" column="roleName" >
                        <generator class="assigned"/>
                </id>

                <property name="keyWords" type="java.lang.String"
              column="keyWords" />

        </class>

</hibernate-mapping>
```

Figure 4.4 Role.hbm.xml : Hibernate Mapping File.

The class *Role* (Figure 4.3) has attributes roleName(String) and keyWords (String). In the mapping file (Figure 4.4), inside the <class> element, the Java class *Role* is mapped to table *Role*. Each attribute of the class *Role* is mapped to a column in the *Role* table. The <id> element defines the mapping between the attributes of class to the primary key column. The value of attribute "column" in the "id" element, in this case *roleName*, is the primary key of the *Role* table.

4.2 Class Structure

4.2.1 Server

We have divided our server component into several packages like **DatabaseAccess, DataTypes, Indexing, Matching,** etc. based on their functionality. Division of code into these

packages enables modularity, reusability, cohesion and de-coupling. The **DatabaseAccess** package has classes like *ReadInput, AccessTopicMapStoredData, AccessUserStoredData,* etc. These classes help to store and fetch the data from the Hibernate based database and provide access to this data to the client side using AXIS2. The **DataTypes** package has classes like *Node, Tool, Topic, Screen, ShortCutScreen,* etc. These are POJO classes which are used to create the basic objects of our application. They are persisted in the relational database with the help of Hibernate. The *Tool* and *Topic* classes extend the *Node* class and the *ShortCutScreen* class extends the *Screen* class. The **Indexing** and **Matching** packages have classes which are used in the information retrieval tool.

4.2.2 Basic Server Working

The first time the Server is started it reads the text file *OMTerms.txt*. The *OMTerms.txt* file is a text file, it stores all the initial information required to setup and create the Topic Maps. A sample *OMTerms.txt* is included in Appendix A. The *GenerateFile* class reads and parses this text file when the server is started for the first time. One line is read at a time using the *LineNumberReader* and *FileReader* classes belonging to java.io package. Every line that is read is used to generate a screen. The line is parsed and is broken into various sub components using various delimiters like ',', ':', '-', '[', ']', '(', ')' , etc. The various sub components that are created using these delimiters are TopicMapId, ScreenLocator, Terms, PathLabels, ToolName, AssociationList. These sub components are used to create objects of the classes like *Node, Tool, Topic* etc. These objects are persisted in the Internal Object Oriented Data Store, by using the methods like save(), update() of *org.hibernate.Session* class. These sub component objects are used to create a *Screen* class object, which represents a screen to the user. Figure 4.5 shows a snapshot of the *OMTerms.txt.*

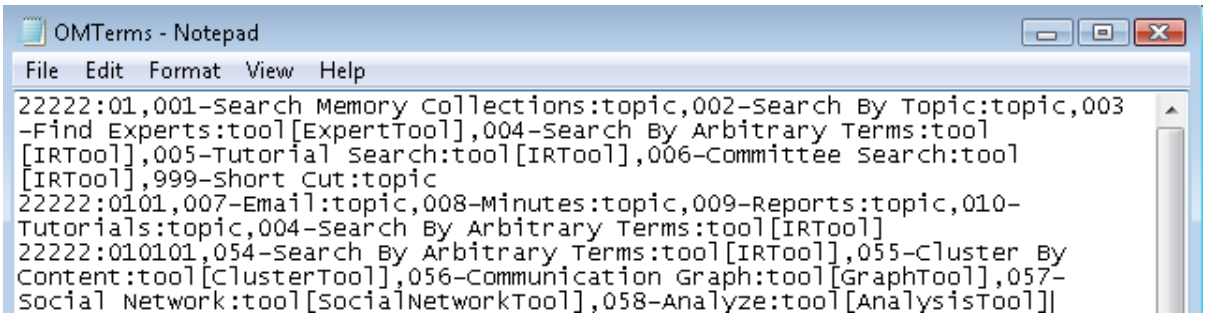

Figure 4.5 Snapshot of *OMTerms.txt.*

4.2.3 Client

The client side is divided into several packages like **DatabaseAccess, DataTypes, DynamicTopicMap** and **Operations** based on their functionality. The **DatabaseAccess** package has classes like *TopicMapData* and *UserData* these classes help to fetch and access the data from the server side using the AXIS2. The **DataTypes** package has classes like *Node, Tool, Topic, Screen, ShortCutScreen* etc the objects of these classes are created using the data fetched from the server side. The **DynamicTopicMap** package has classes like *GenerateTopicMap, GraphicsPanelFactory, SetupScreen* etc. These classes help in rendering and creating the topic map. For rendering screens of the Topic Maps, we use several Java Swing components such as JPanel, JFrame etc. The **Operations** package has several classes like *CreateNode, GeneralizeANode, ModifyANode, SpecifyANode, ChangePathLabel* etc. Objects of these classes help to perform various operations in the Group Topic Maps.

4.3 AXIS2 Based Client - Server Communication

Our prototype runs on the client side and communicates with the server using the SOAP protocol. SOAP version 1.2 is a lightweight protocol for exchange of information in a decentralized, distributed and heterogeneous environment. It is an XML based protocol that consists of four parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, a convention for representing remote procedure calls and responses and a binding convention for exchanging messages using an underlying protocol [29].

4.3.1 Server Side

In our implementation we have used Apache AXIS 2. Apache AXIS 2 is an implementation of SOAP and REST [30]. The Eclipse plug-in wizards: Axis2 Code Generator and Axis2 Service Archiver are used to generate the WSDL and the Web Service Archive (.aar). WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services) [31].

The Web Service Archive handles and services the Client request. The Server Java Code which provides interface to the Internal Object Oriented Data Store (Hibernate Database) is used as the input to the Axis2 Service Archiver to generate the Web Service Archive (.aar). The generated .aar is placed in the "catalina\webapps\axis2\WEB-INF\services" folder and the server is started to provide the services offered by the server.

The server also provides access to the Information Retrieval tool. The Information Retrieval tool is also AXIS2 based. The Information Retrieval tool is used to search the Organizational Repository and fetch documents, presentations, emails etc corresponding to the keywords provided by the client. The resources fetched from the Information Retrieval tool are displayed to the client in the form of JavaScript enabled web page. These web pages are created dynamically corresponding to the current request. The content of the various documents that are fetched can be contracted and expanded according to the user requirement. This information retrieval tool was developed by our research group and was integrated and modified by me to meet our requirements.

4.3.2 Client Side

The RPCServiceClient object provides client access to the service. The EndpointReference object is used as a remote reference to a web service endpoint. The QName object contains a Namespace URI of the operation. RPCServiceClient class object is created, it is set to point to the endpoint reference, it's invokeBlocking() method is invoked passing the qname, arguments of the operation and the return type of the result. Figure 4.6 shows the sample AXIS2 client code.

```
RPCServiceClient serviceClient = new RPCServiceClient();
Options options = serviceClient.getOptions();
EndpointReference targetEPR = new
EndpointReference("http://129.186.93.177:8080/axis2/services/NewService");
options.setTo(targetEPR);
QName opFindBaseNode =new QName("http://databaseAccess", "findBaseNode","ns");
Object[] opFindBaseNodeArgs = new Object[] {sourceScreenLocator };
Class[] opFindBaseNodeTypes = new Class[] { String.class  };
Object[] opFindBaseNodeResponse =serviceClient.invokeBlocking(opFindBaseNode,
opFindBaseNodeArgs,opFindBaseNodeTypes);
String opFindBaseNodeResult = (String) opFindBaseNodeResponse[0];
```

Figure 4.6 AXIS2 Client Code.

4.4 Swing Based User Interface

The end user is provided with the interactive Swing based Graphical User Interface. Our implementation provides users with access to multiple Process and Group Topic Maps. The user's logon id and password are used to determine the list of topic maps that a user can access. Figure 4.7 shows the initial screen of a Process Topic Map based on the Federal Reserve Monetary Policy process.

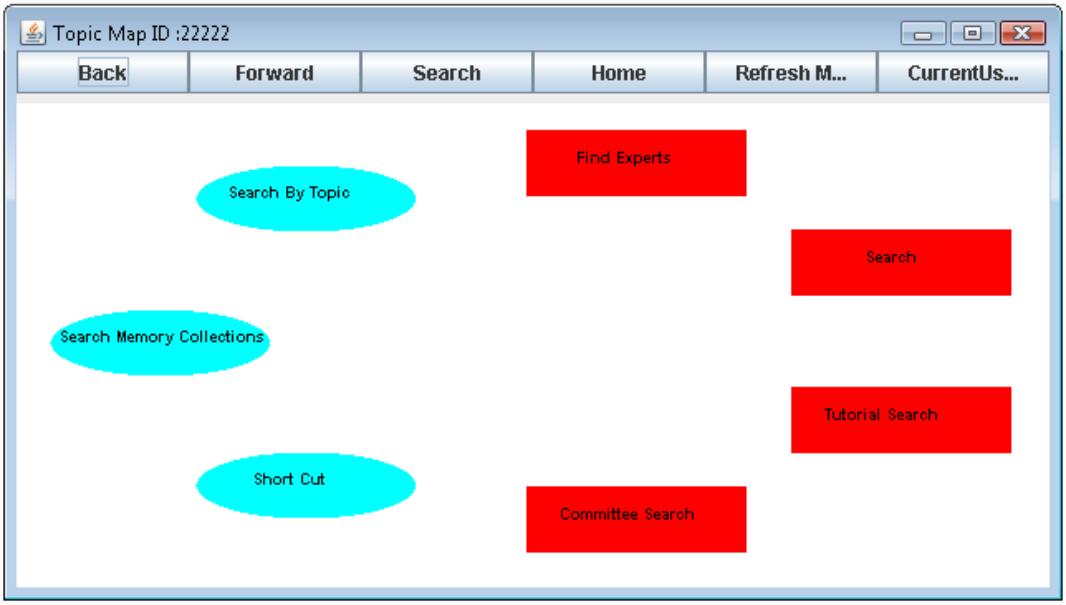

Figure 4.7 Initial screen of a Topic Map based on the Federal Reserve Monetary Policy process.

In our implementation a visual topic map is represented by set of screens. A recursive screen structure is created using Java Swing's JPanel, JFrame etc. Each screen has some nodes associated with it. The nodes in a topic map are displayed as either ovals or rectangles. The rectangles represent tools and the ovals represent topic nodes containing topics. The tools and topics can occur at any level in the topic map. Navigation in the topic map is flexible and can be controlled either by clicking on a topic map node (oval or rectangle) or using the level navigation icons at the top of the screen. When, we click on a topic we are directed to another screen which has relevant topics. This continues recursively until we reach the leaf nodes. With leaf topic nodes several resources are associated and with leaf tool nodes tools for Information Retrieval, Frequency Analysis are associated to fetch occurrences. The Information Retrieval tool searches the Organizational Repository for documents and other information sources, using the screen terms of the topics traversed till now as the keyword. The back button at the top allows the user to move back one level in the topic map, and the forward button allows the user to move forward in the topic map. The actual forward move depends on the

user's previous operation. If the previous operation was the result of clicking either the home or the back arrow, clicking on the forward arrow will return the user to her/his previous location in the topic map. The Home Button at the top is used to reach to the home screen (i.e the first screen) from any screen the user was present. As expected, using the back and home buttons strips the term(s) from the path label and the use of forward button append term(s) to the path label. So, that it is correct for the resulting topic map level. The user can also use the Information Retrieval tool by clicking the Search button at the top of the screen, screen terms corresponding to all the nodes traversed till now are used as the keywords for input. The Current User button is used to determine the Users that are currently logged in and are using the system. Moreover, after performing any operation, the refresh button can be used to refresh the topic map and show the latest state of the Topic Map. The node icon Short Cut presented on the first screen points to the Short Cut screen, Figure 4.8 depicts the Short Cut Screen. In this example the user has copied the Cluster By Content node to the Short Cut screen. The user can personalize the Short Cut screen by adding any nodes from deeper levels in the topic map. The user can clean up the shortcut screen by deleting nodes. The copy is a shallow copy that simply points to the node. The user can open a window for each of the topic maps that they have access to. For operations such as DeleteANode, ModifyANode, MoveANode, and DuplicateANode, the user operates directly on window that represents the Group Topic Map. He/she simply clicks on the desired node and selects the appropriate operation from the menu.

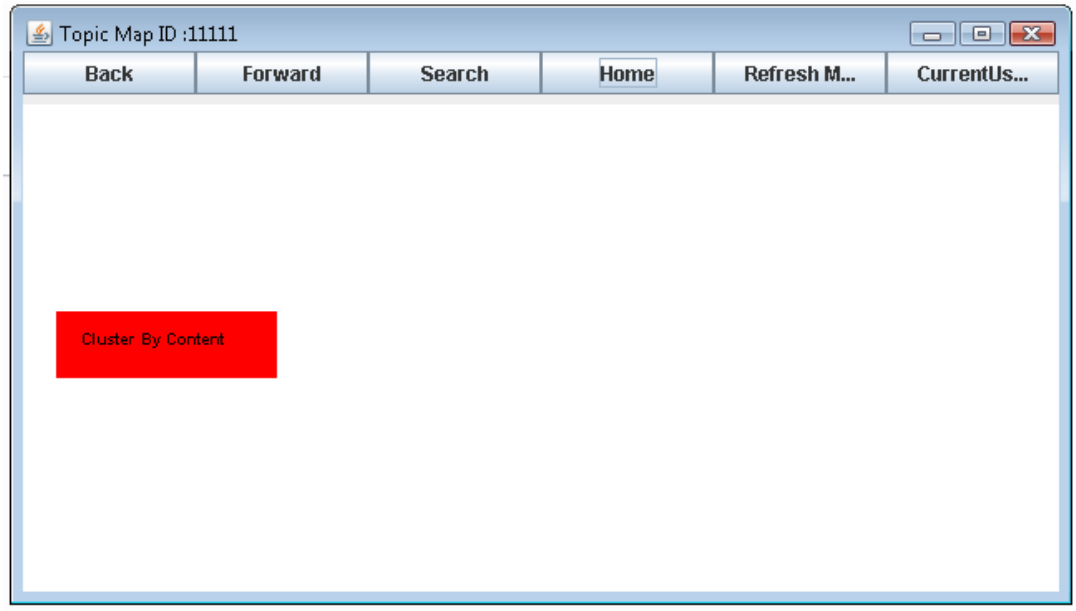

Figure 4.8 Shortcut Screen.

4.5 Group Topic Map Operations

Before any operation modifies the topic map, it is checked whether the topic map is modifiable or not. The getEditable() method is invoked at the server side using the AXIS2. The getEditable() method is invoked using the parameter topicmapid. This method queries the database to fetch the edit details of a topic map, using topicmapid in the query. For the group topic map, the method returns true and for a process topic map it returns false. If the method returns true then the operation is performed otherwise it is aborted.

Moreover, before any operation is performed, the current version of the Group Topic map the user has access to is compared to the latest version stored in the database. The latest version from the database is obtained by invoking the method getVersion() on the server by using AXIS 2. The topicMapId is passed as the argument to the getVersion() method. If the version matches then the operation is performed and a new version number is assigned to that topic map. If the versions don't match then the topic map is refreshed and the latest topic map is rendered to the user and then the user can perform the required operation. In the remainder of this section we look at the implementation of individual operations.

4.5.1 CreateANode

This operation uses the *CreateANode* class belonging to **Operation** package. This class extends the JFrame class. A panel is created using JPanel. A radio button for selecting either Tool or Topic is added to this panel. The inner class RadioButtonHandler which implements the ItemListener, listens to the selection of radio buttons.

If the radio button for Tool is selected, it creates a *CreateTool* class object. The *CreateTool* class extends JFrame class, this class invokes the getAvailableToolList() method at the server side, using AXIS2 . This method returns the list of tools that are currently active and supported by the server. Dynamically an array of radio buttons is created using the list of tools just retrieved. An object of RadioButtonHandler class which implements ItemListener is created to listen the event generated by selecting a radio button, its itemStateChanged() method is used to determine the radio button selected. A submit button is provided, an object of inner class ButtonHandler which implements ActionListener class is used to listen to this submit button, its actionPerformed() method, determines the tool to be generated and it invokes createTool() method at the server side, using AXIS2. The arguments topicMapId, screenLocator, pathLabel, screenTerm and toolName are used as parameters invoke createTool() method.

If the radio button for Topic is selected, it creates a *CreateTopic* class object. The *CreateTopic* class extends the JFrame class, this class invokes the getAvailableResourceList() method at the server side, using AXIS2 . This method returns the list of resources that are currently active and supported by the server. Dynamically an array of check boxes is created using the list of resources just retrieved. An object of inner class CheckBoxHandler class which implements ItemListener is created to listen the event generated by selecting the check boxes, its itemStateChanged() method is used to determine the check boxes selected. A submit button is provided, an object of inner class ButtonHandler which implements ActionListener class is used to listen to this submit button. Its actionPerformed() method, determines the resources to be associated with the new topic and it invokes createTopic() method at the server side, using AXIS2. The arguments topicMapId, screenLocator, pathLabel, screenTerm and associationList are used as parameters to invoke createTopic() method.

4.5.2 ModifyANode

This operation uses the *ModifyANode* class belonging to the **Operation** package. This class extends the JFrame class. A panel is created using JPanel. A radio button for selecting changeName, changePathlabel or changeResourceTool is added to this panel. The inner class RadioButtonHandler implements the ItemListener, it listens to the selection of radio button. Its itemStateChanged() method is used to determine which of the radio buttons was selected.

If the radio button for changeName is selected, it creates a *ChangeName* class object. *ChangeName* class is present in the **Operation** package and extends JFrame class. A panel is created using JPanel. A text field and a submit button are added to this panel. An object of inner class ButtonHandler which implements ActionListener class is used to listen to this submit button. Its actionPerformed() method, gets the new screenTerm and it invokes the changeNodeName() method at the server side, using AXIS2. The arguments topicMapId, targetScreenLocator, index, topicNo, nextSourceScreenLocator and screenTerm are used as parameters to changeNodeName() method.

If the radio button for changePathLabel is selected, it creates the *ChangePathLabel* class object. ChangePathLabelclass is present in the Operation package and extends JFrame class. A panel is created using JPanel. A text field and a submit button are added to this panel. An object of inner class ButtonHandler which implements ActionListener class is used to listen to this submit button. Its actionPerformed(), gets the new pathLabel and it invokes changePathLabel() method at the server side, using AXIS2. The arguments topicMapId, targetScreenLocator, index,sTerm,topicNo and nextSourceScreenLocator are used as parameter changePathLabel() method.

If the radio button for changeResourceTool is selected, it determines whether the node is a topic, tool or leaf topic node. If a node is a non-leaf topic, then the operation is not completed. In our current model we can change only the Tool or Resource associated with the leaf node. Depending upon whether we have Topic or Tool, different objects of *ChangeResourceTool* class are created, using different constructors. The same logic and concept as used in the *CreateANode* class is used, as affectively in the modify node a new node is created and the old nodes are deleted.

4.5.3 GeneralizeANode

This operation uses the *GeneralizeANode* class belonging to **Operation** package. This class extends the JFrame class. A panel is created using JPanel. A text field and submit button is added to this panel. The object of inner class ButtonHandler which implements the ActionListener listens to the submission button. Its actionPerformed() method is used to determine to the new search terms, these search terms are combined with current the path label. The actionPerformed() method then invokes the specifyOrGeneralizeANode() method at the server side, using AXIS2. The specifyOrGeneralizeANode() method is invoked using the arguements topicMapId, screenLocator, finalPathLabel, screenTerm, nodeType. This method creates a new node on the same screen as the original node. It stores the new node details like topicmapid, screenlocator,the new path label and associates the Information Retrieval Tool with the new node.

4.5.4 SpecifyANode

This operation uses the *SpecifyANode* class belonging to the **Operation** package. This class extends the JFrame class. A panel is created using JPanel. A text field and submit button is added to this panel. The object of inner class ButtonHandler which implements the ActionListener listens to the submission button, its actionPerformed() method is used to determine to the new search terms, these search terms are combined with current the path label. The actionPerformed() method then invokes getKeyWords() method at the server side, using AXIS2. It is used to retrieve the keywords specific to a role, using topicmapid and userid as the parameters. It also invokes the specifyOrGeneralizeANode() method at the server side, using AXIS2, to generate the new node.

4.5.5 AddThisNode

This operation uses the *DataForCopyNode* class belonging to the **Operation** package. This class has static attributes like sourceScreenLocator, sourcePathlabel, nodetype, nodePathLabel, nextSourceScreenLocator, screenTerm. These attributes store the information of the node that would be copied. The user then navigates to the Group Topic Map and the screen where he/she wants to add

this node. The user then right clicks on the empty space and from the popup menu selects the CopyANode operation. The target topic map and screen information is stored in the static attributes like targetScreenLocator, targetPathlabel, targetNodePathLabel, targetNextSourceScreenLocator of the DataForCopyNode class. The CopyANode operation invokes copyNode() method on the server using AXIS2. The copyNode() method is invoked using the static attributes of the class DataForCopyNode as the arguments. The copyNode() method recursively combines the copied node's details and it's next screen linkages details with the topic map id and the screen locator of the destination Group Topic map and Screen and stores this information in the database. Moreover when a node is being copied, it is checked whether it a leaf topic node, tool node or a non leaf topic node. This process continues recursively till the leaf nodes are processed.

4.5.6 CopyToShortCutScreen

The CopyToShortCutScreen is similar to the AddThisNode operation. But in case of the CopyToShortCutScreen the target or the destination screen is the ShortCutScreen. Thus the targetScreenLocator, targetPathlabel, targetNodePathLabel, targetNextSourceScreenLocator of the *DataForCopyNode* class are always set to point to ShortCutScreen.

4.5.7 DeleteANode

This operation uses the *DataForCopyNode* class belonging to the **Operation** package. This class has static attributes like sourceScreenLocator, sourcePathlabel, nodetype, nodePathLabel, nextSourceScreenLocator, screenTerm. These attributes store the information of the node that would be deleted. The deleteNode() method is invoked on the server using Axis2. The static attributes sourceScreenLocator, sourcePathlabel, nodetype, nodePathLabel, nextSourceScreenLocator, screenTerm of the DataForCopyNode class are passed as the argument to the deleteNode() method. The deleteNode() method removes the specific node detail, this process continues recursively till the leaf nodes are processed.

4.5.8 MoveANode

In the MoveANode the operation AddThisNode is first performed followed by the DeleteANode operation. The details of these operations have been discussed above.

4.6 Web based administrative tool.

A web based administrative tools is provided. This tool is used by the Topic Map Administrator to perform administrative operations. The web based administrative tool helps in granting the user access to specific topic maps, defining different user roles, providing metadata for a specific role, maintaining the list of available resources and tools, entering and modifying topic map details. In our implementation Servlets are used for providing the administrator with the graphical user interface. Hibernate is used to store, access and manipulate these details in the MySql relational database. Apache Tomcat is used as the Servlet Container for the Servlets.

When the administrator selects the "Edit Topic Map" operation then he/she is directed to another page where he/she can select "Delete or Modify Topic Map Detail" or "Enter New Topic Map Details" by clicking appropriate button. When the user selects the "Delete or Modify Topic Map Detail" button , then the user is provided with the list of all the topic maps whose details have been entered, the user can either delete or modify these details, by modifying the content of the text fields and by checking unchecking the check boxes. When the user selects the "Enter New Topic Map Details", the user is provided with a text field to enter the topic map name, radio buttons are provided to mark the topic map as the process or group topic maps. If sufficient details are not provided then the user is requested to provide the required detail. When the user clicks the submit button these changes are persisted in the Hibernate database.

Similarly, for other operations like "Edit Role", "Edit User", "Edit Tool List", "Edit Resource List" the user can select to delete or modify the existing details by modifying the content of a text field and by checking or unchecking the check boxes. The user can also provide new details like in "Edit Role" operation the administrator provides the role name and the metadata he/she wants to associate with this role in the text boxes. In "Edit User" the administrator provides the new username and the password to the user, the administrator also assigns the topic maps and the roles to the user. In "Edit Tool List" and "Edit Resource List" operation administrator enters the new tool and resource name.

4.7 Example Scenario

In this section we discuss an example scenario and show the screens that are generated during different operations.

4.7.1 User Login

Each user uses his username and password to login. This username and password is assigned to him by the topic map administrator. Multiple users may be logged into the system at a given time. In our example scenario, the user “alok” signs in to the system. Figure 4.9 shows the login prompt displayed to the user.

Figure 4.9 Login Prompt.

When the user logins, on the first screen all the Topic Maps – Process and Group Topic Maps he is registered are displayed. A User can access and use multiple Topic Maps simultaneously. When the user clicks a particular Topic Map node, that Topic Map is started. The user “alok” has access to three topic maps “11111”, “22222” and “33333”, these are displayed on the first screen, when the user “alok” logins. Figure 4.10 shows Topic Maps available to user “alok”.

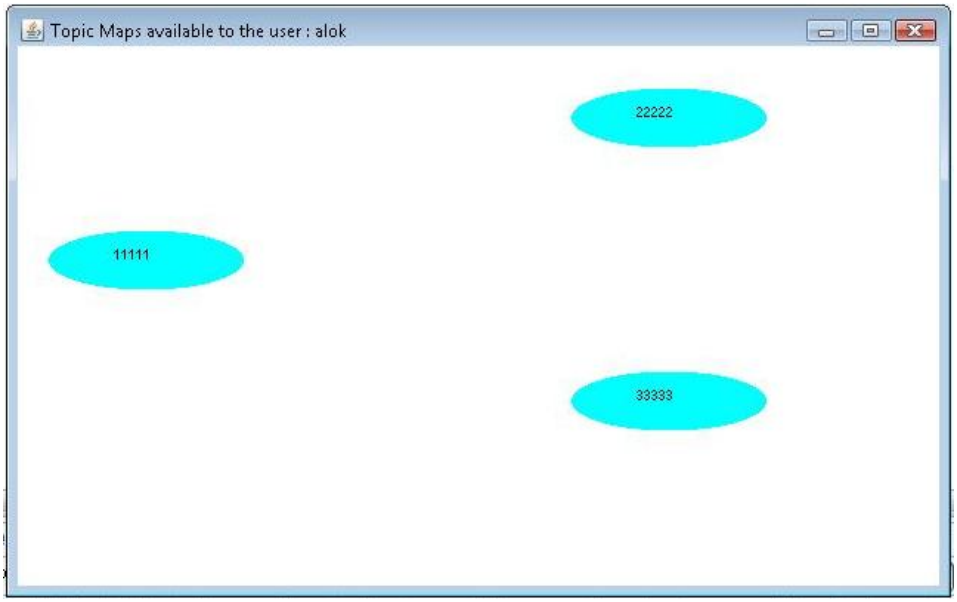

Figure 4.10 Topic Maps available to the user “alok”.

If the Topic Map is a Group Topic Map the user would be able to modify it according to his/her understanding. In case of the Process Topic Maps they are static they can’t be modified by the

user as they are represent the Organization wide process. In our example scenario the Topic Map "11111" is a Process Topic Map, the user can't modify it. When the user tries to modify it, the user gets the message "Topic map is uneditable". The user clicks "OK" and then the user returns to the Topic Map. Figure 4.11 shows process topic map which can't be modified.

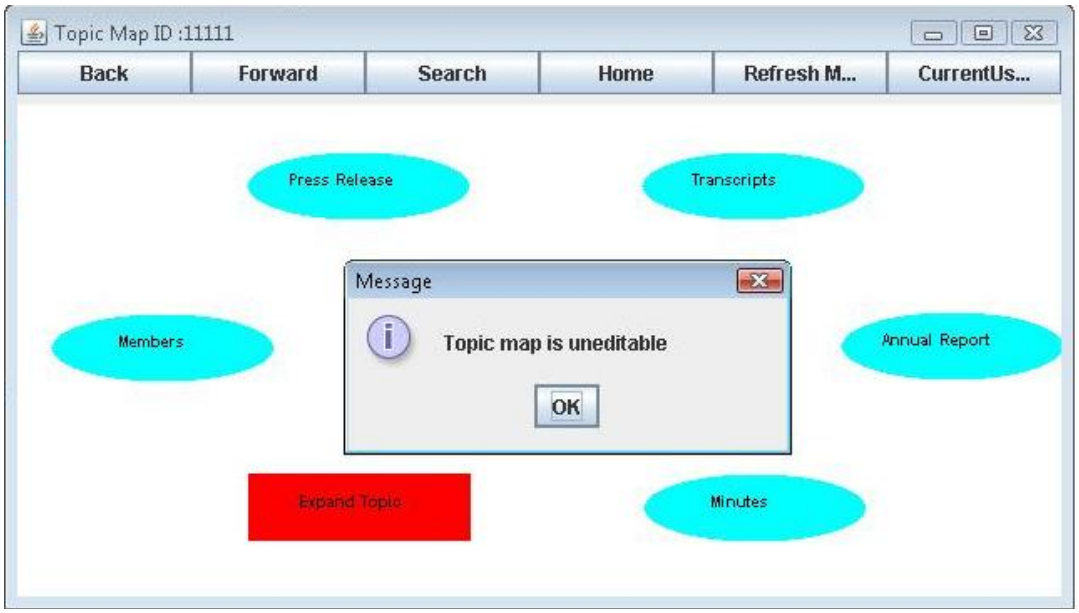

Figure 4.11 Topic Map "11111" is non-editable.

The user can traverse the topic map by clicking on the topics, thus being directed from one screen to other screen or can use the tools or resources associated with the node. The user can perform operations on the Group Topic Map nodes by right clicking a node and then choosing the operation from the menu list. The operations SpecifyANode and GeneralizeANode are not allowed for leaf nodes in the current implementation.

4.7.2 AddThisNode

The user navigates the process topic map and right clicks on the specific node he wants to add from process topic map to a group topic map. The Figure 4.12 shows user navigating the process topic map "11111", right clicking on the "Press Release" node and then selecting "AddThisNode" operation.

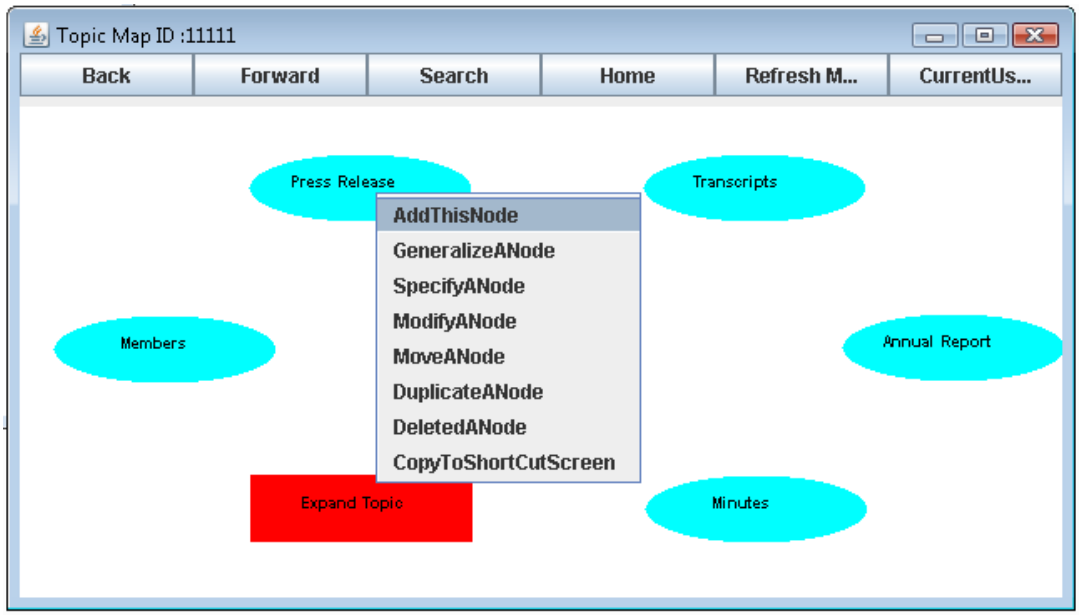

Figure 4.12 User Selects the AddThisNode Operation.

The user then navigates the desired Group Topic Map and on the required screen right clicks on the empty space at screen and selects the "CopyTheNode" operation from the menu. Figure 4.13 shows user navigating Group Topic Map "22222", right clicking and selecting the "CopyTheNode" operation.

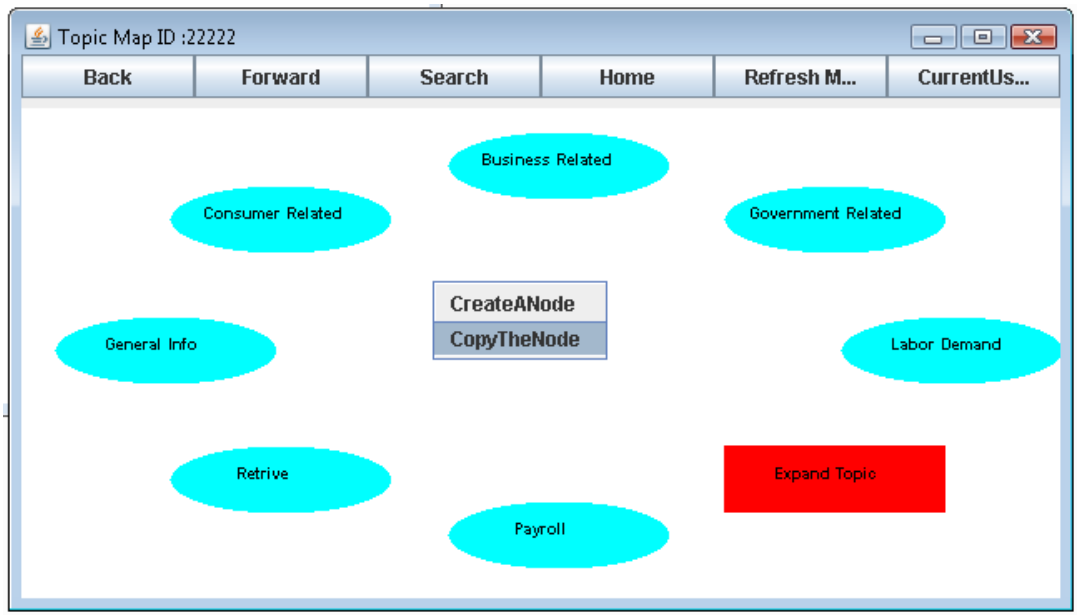

Figure 4.13 User selects the CopyTheNode operation.

Figure 4.14 shows the node “Press Release” copied to the Group Topic Map “22222”, after the user selects the CopyTheNode operation and presses the refresh button.

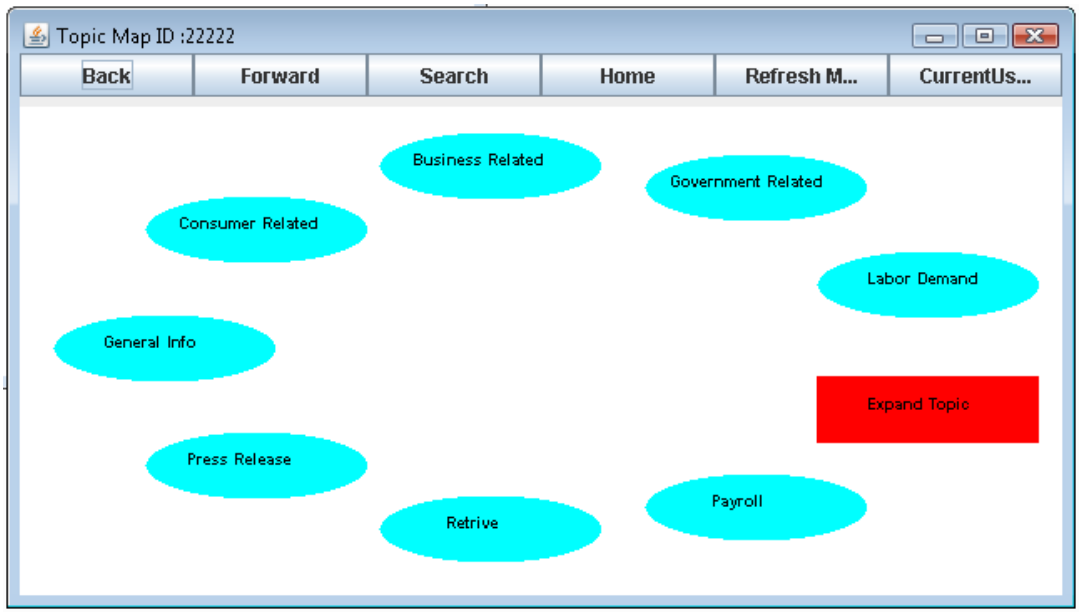

Figure 4.14 Topic Press Release copied to the screen.

4.7.3 CreateANode

The user right clicks on the screen where he wants to create a new node. Then the user selects the CreateANode operation from the menu as shown in Figure 4.15.

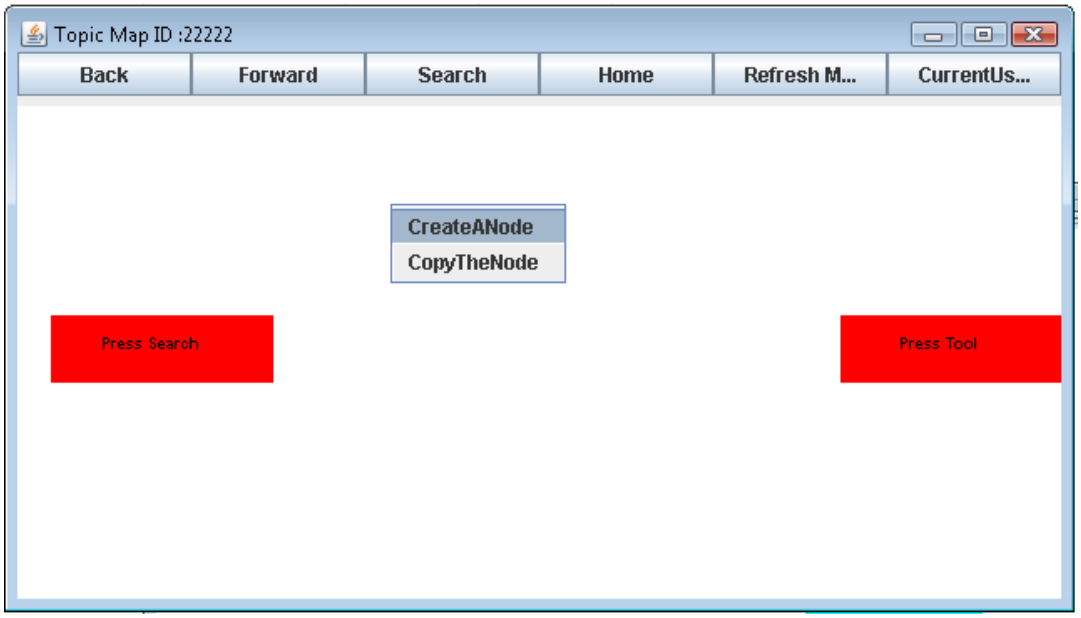

Figure 4.15 User selects the “CreateANode” operation.

The user can create the new node as a Topic or Tool. The user can choose the node type using the radio button as shown in the Figure 4.16.

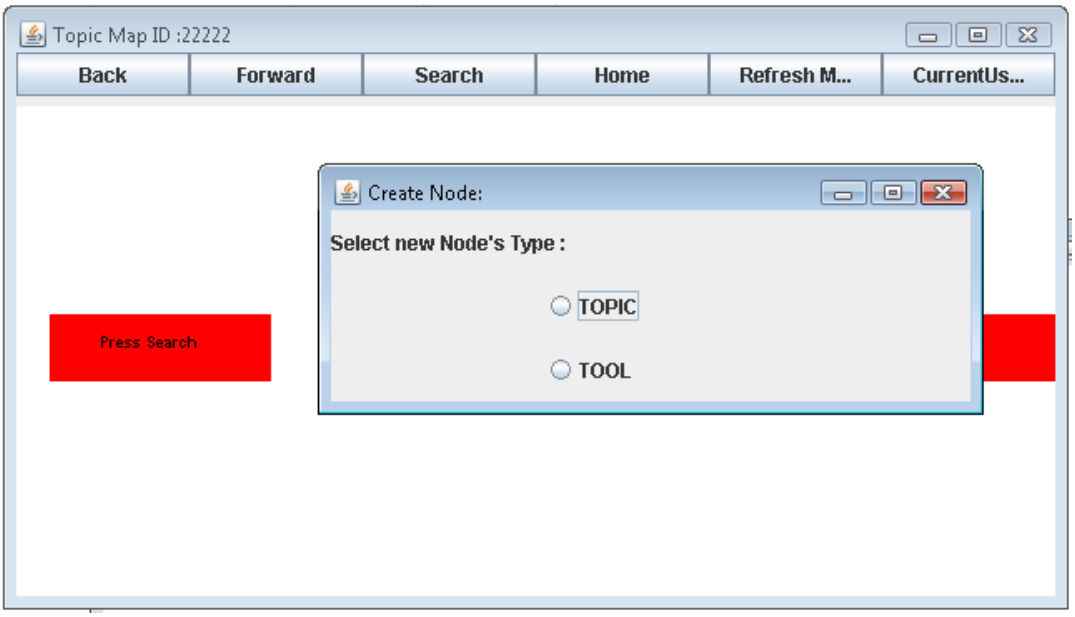

Figure 4.16 User given option to create Tool or Topic.

When the user selects create a new Tool, then the user can provide the new Tool Name and selects the Tool Type the user want to associate the new node as shown in Figure 4.17.

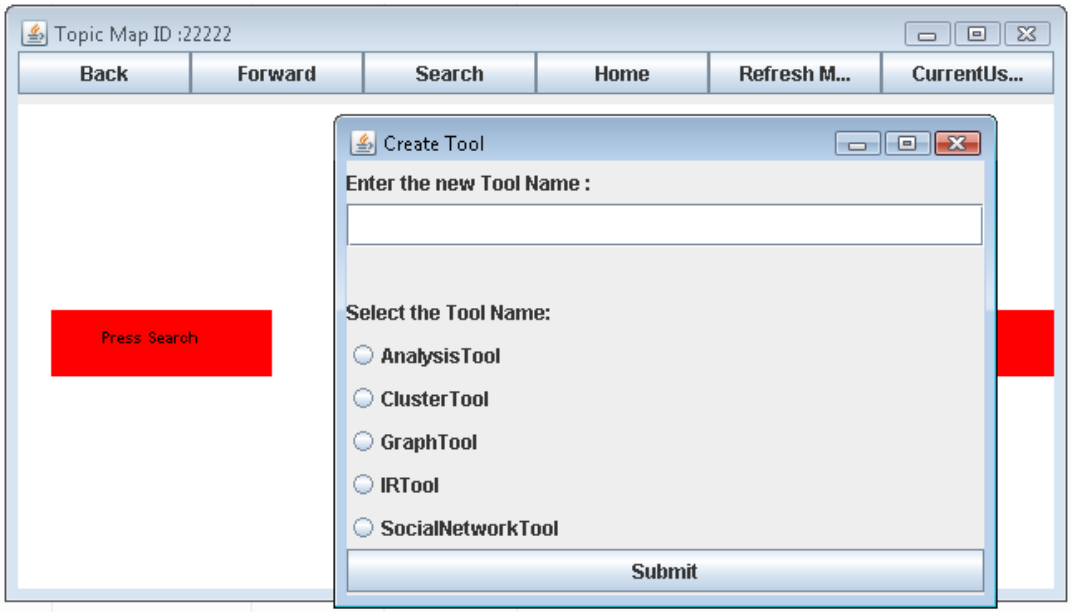

Figure 4.17 User has to enter the Screen term and Tool type.

In our example scenario we create a new node with the screen term “New Tool” and associate the Analysis Tool with this node, as shown in Figure 4.18.

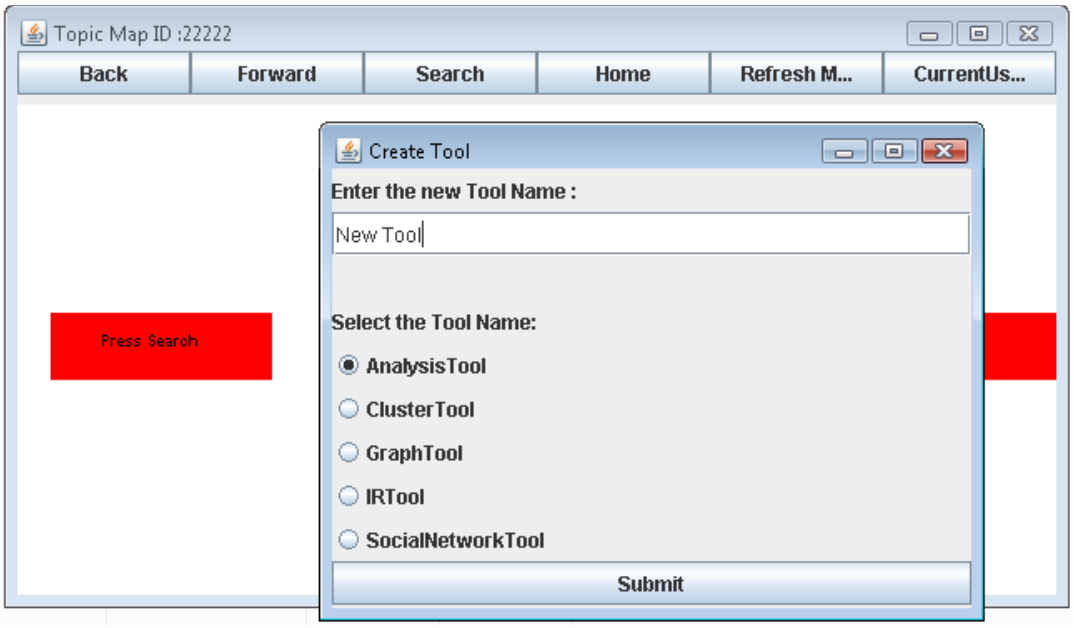

Figure 4.18 User provides Screen term and selects the Tool type.

When the user clicks the Refresh, the new Tool appears on the screen, as shown in the Figure 4.19.

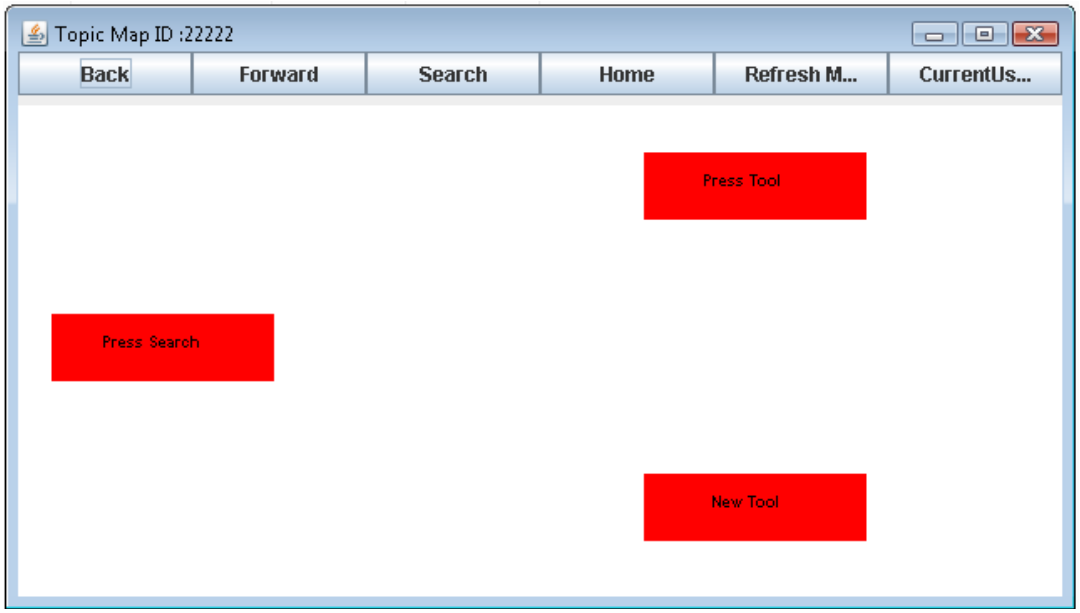

Figure 4.19 New Tool appears on the screen.

If the user selects to create a new topic, then the user provides the new topic name and select the resources he wants to associate with this topic node, as shown in Figure 4.20.

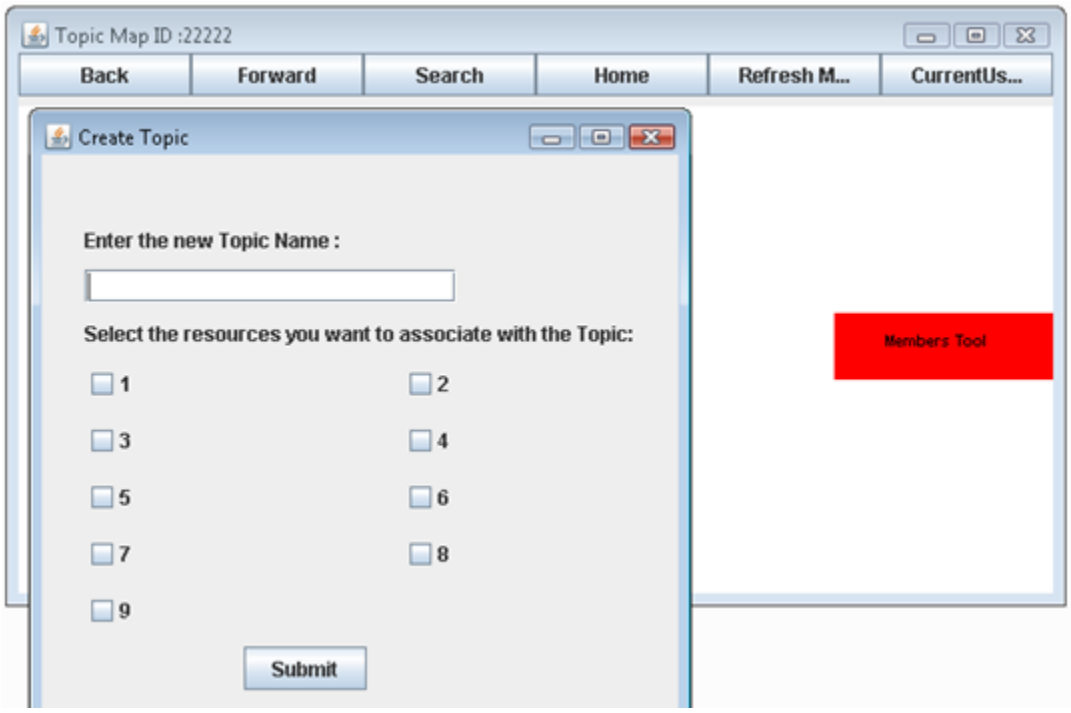

Figure 4.20 User has to enter the Screen term and select the resources.

In our example scenario we create a new Topic node with screen term “Memory Resources” and we associate the resources 3, 5 and 6 with this topic node as shown in Figure 4.21.

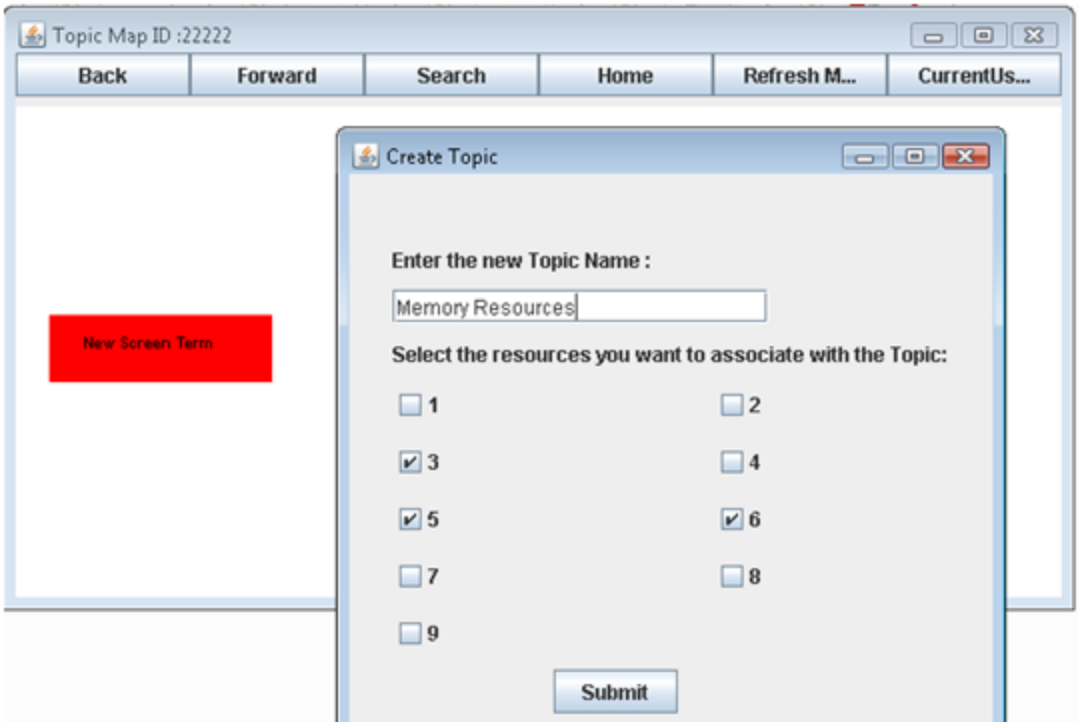

Figure 4.21 User provides Screen term and selects the resources.

When the user clicks the Submit button and then clicks the Refresh, the new topic "Memory Resources" appears on the screen, as shown in the Figure 4.22.

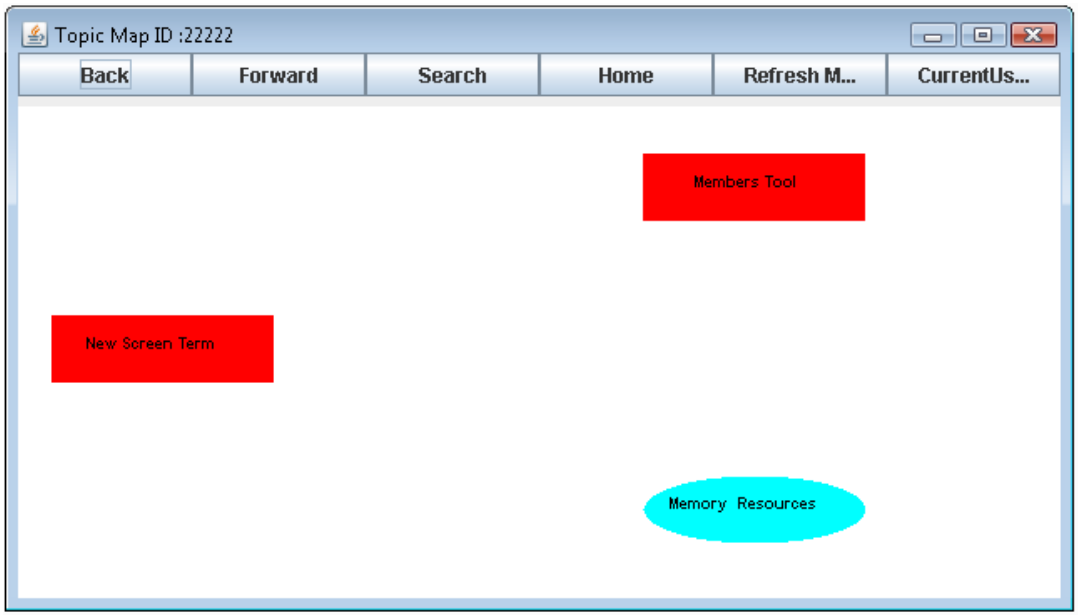

Figure 4.22 New topic "Memory Resource" is created.

4.7.4 GeneralizeANode

The user right clicks on the node he wants to Generalize. Then the user selects the GeneralizeANode operation from the menu, as shown in the Figure 4.23, the user right clicks on the node “Transcripts” and then selects the GeneralizeANode operation.

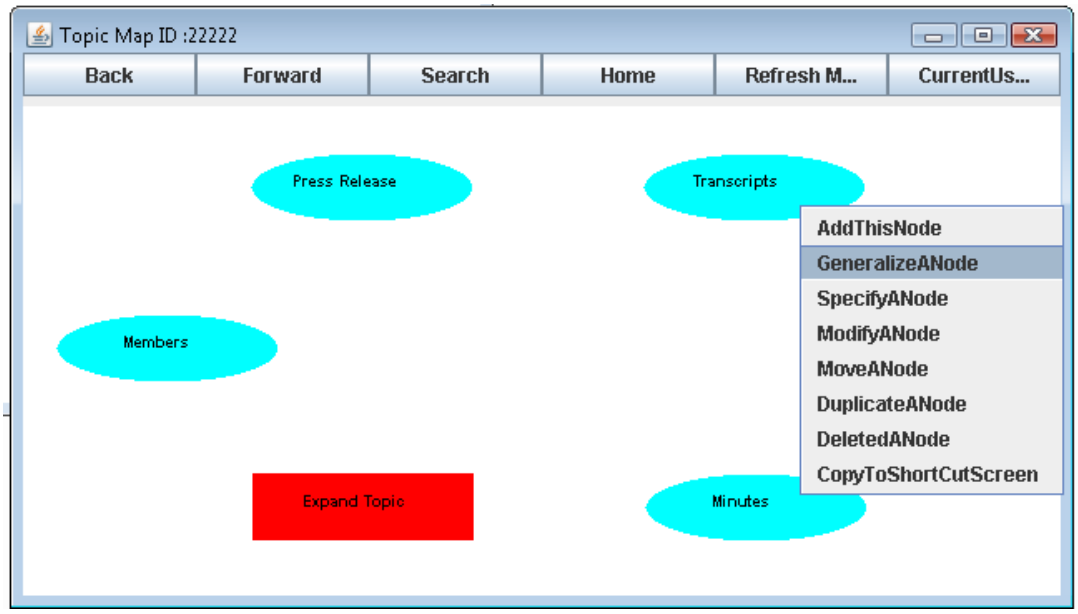

Figure 4.23User selects the GeneralizeANode operation.

The user is provided with a text field to enter the Additional Search Terms, as shown in Figure 4.24.

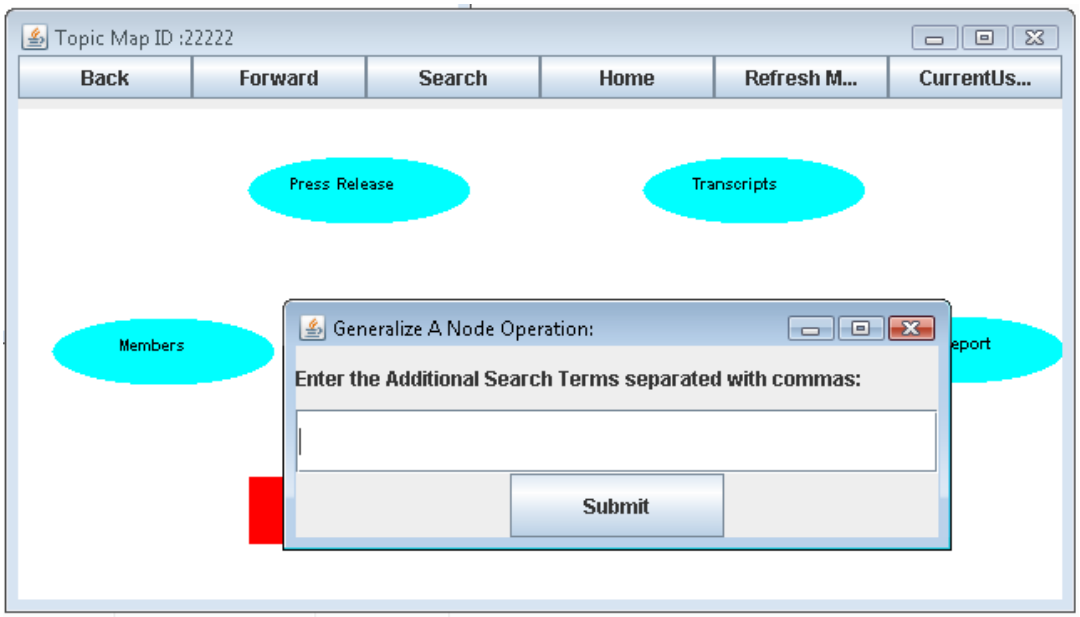

Figure 4.24 Text Field to enter additional search terms.

The user provides the additional search terms “emails, recording, messages” in the text box and presses the submit button as shown in Figure 4.25.

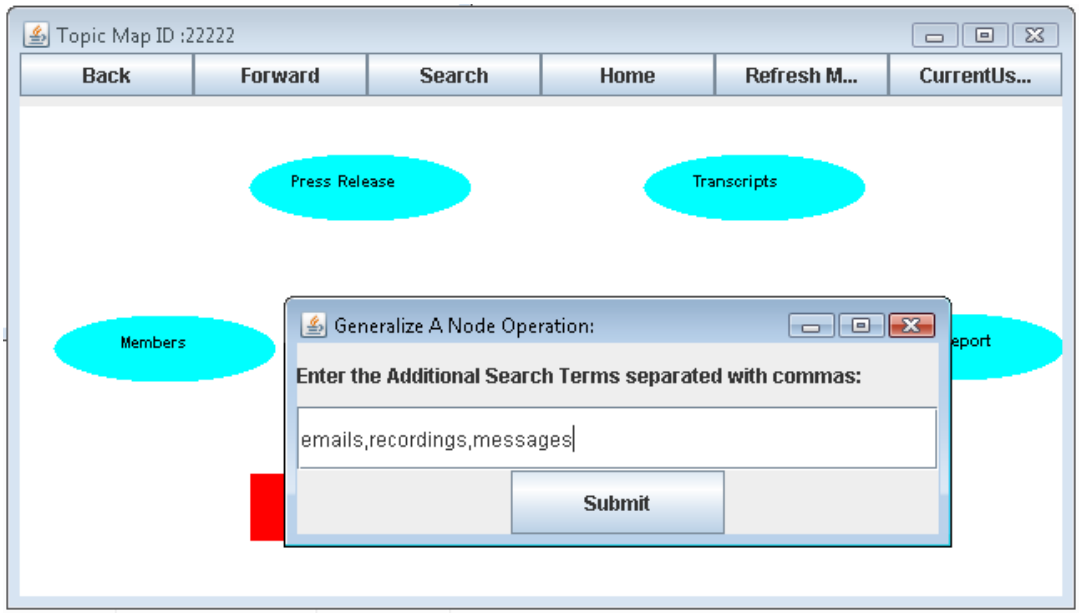

Figure 4.25 User enters additional search terms.

When the user clicks the Refresh, the changes are preserved and the new node “Generalized Transcripts” appear on the screen as shown in the Figure 4.26.

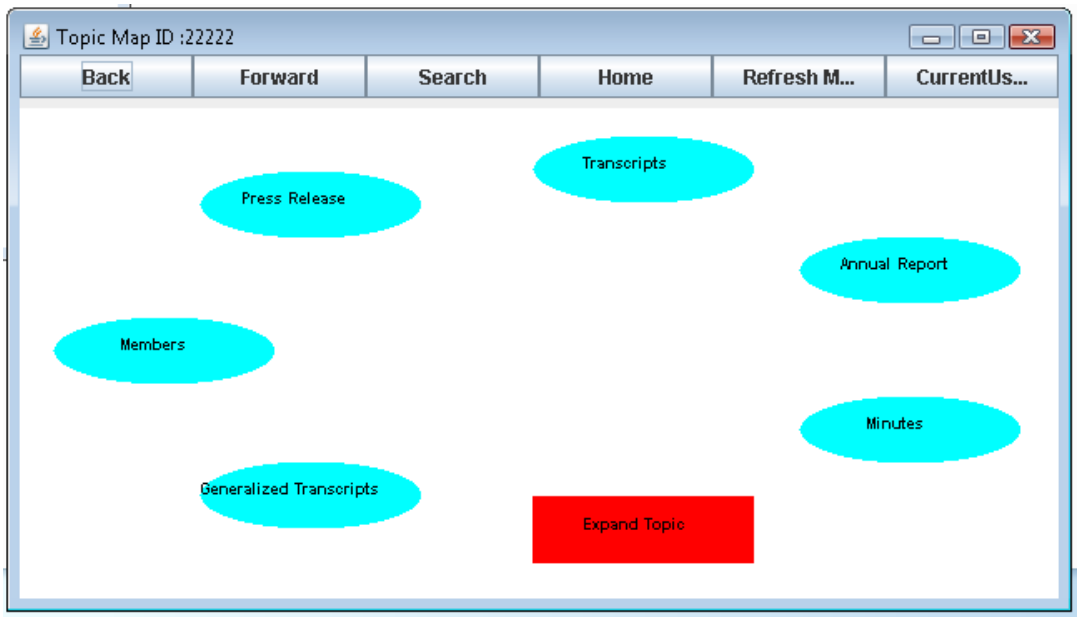

Figure 4.26 “Generalized Transcripts” node created on screen.

4.7.5 SpecifyANode

The user right clicks on the Payroll Tool node. Then the user selects the GeneralizeANode operation from the pop menu as shown in the Figure 4.27.

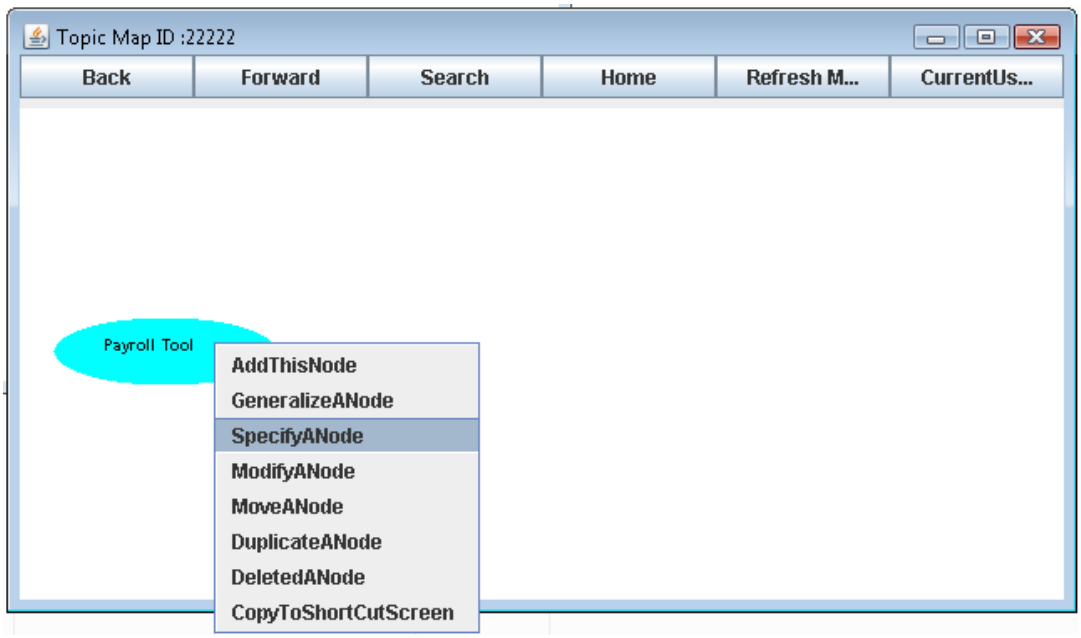

Figure 4.27 User selects the SpecifyANode operation.

Then the user is provided with the text filed to enter the additional search terms separated by commas, as shown in Figure 4.28.

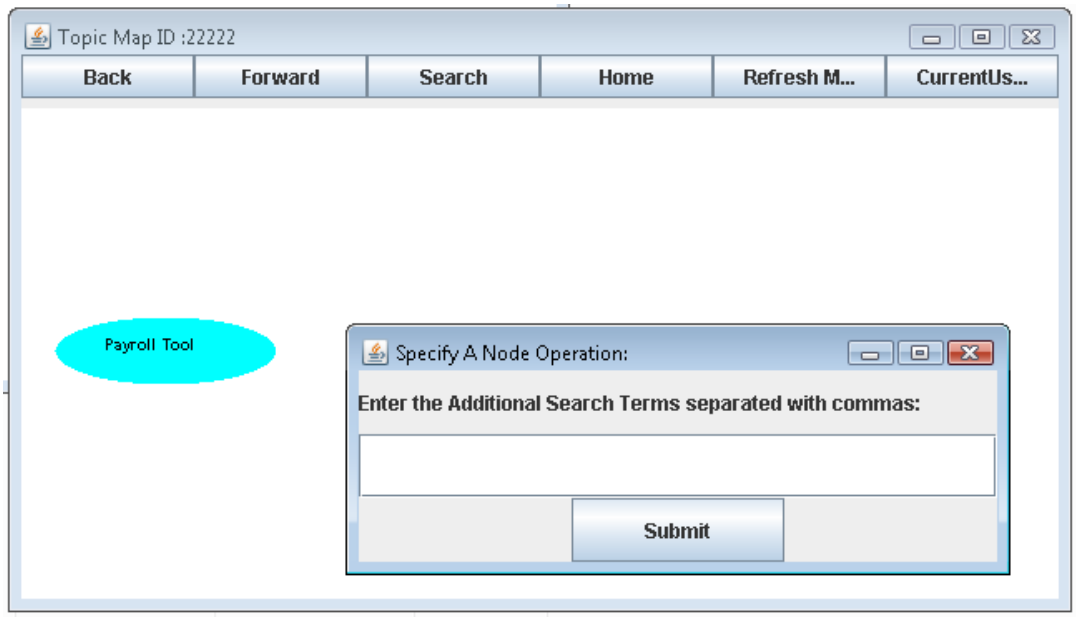

Figure 4.28 Text Field to enter additional search terms.

The user provides the additional search terms “salary, compensation, allowances” in the text field and clicks the submit button as shown in the Figure 4.29.

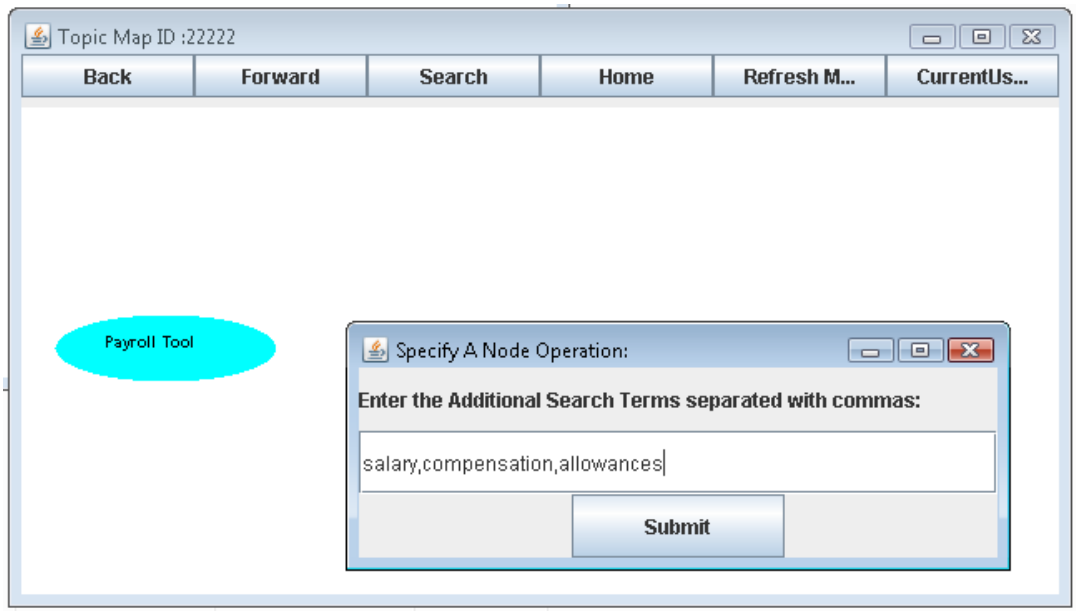

Figure 4.29 User enters additional search terms.

When the user clicks the Refresh, the changes are preserved and the new node “Specified Payroll Tool” appears on the screen as shown in the Figure 4.30.

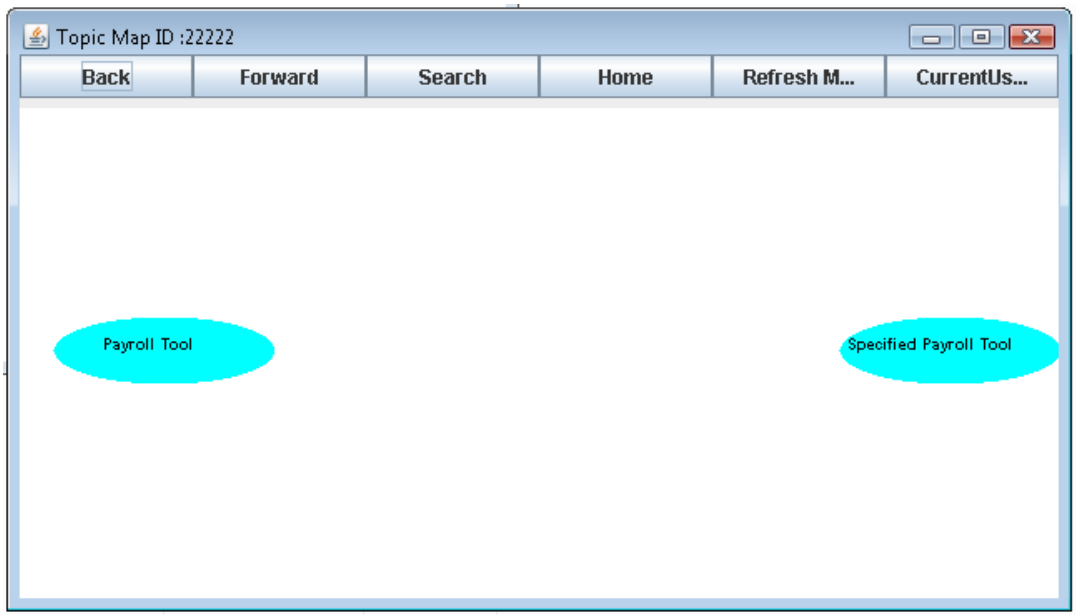

Figure 4.30 “Specified Payroll Tool” node created on screen.

4.7.6 ModifyANode

The user right clicks on the "Member Search" node as shown in the Figure 4.31. Then the user selects the ModifyANode operation from the menu.

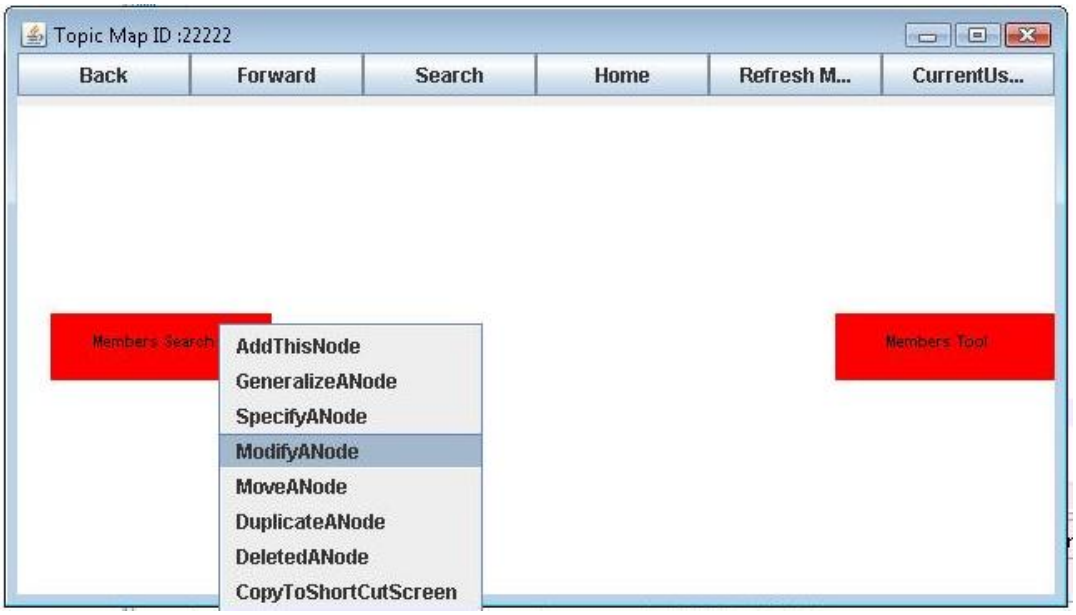

Figure 4.31 User selects the ModifyANode operation.

The user can perform three modification operations: Change the Name, Change the PathLabel, Change the Resource or Tool Type associated with the node as shown in Figure 4.32. The operations Change the Resource or Tool Type can be only performed with the Leaf Nodes.

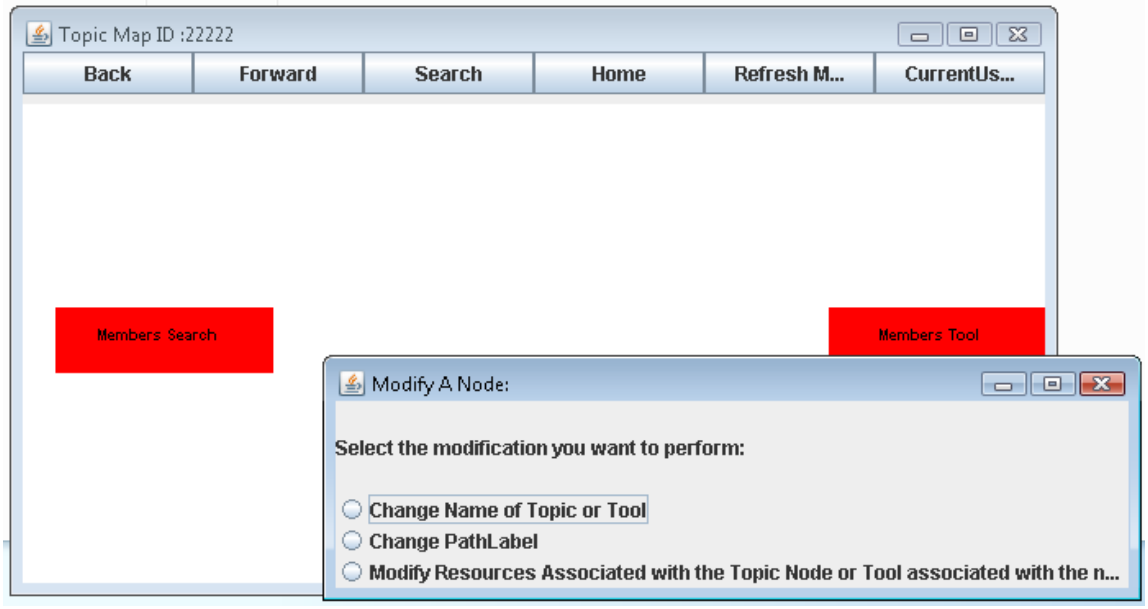

Figure 4.32 Different modification option provided to the user.

When the user selects Change the Name, then the user can provide the new screen name for the topic or the tool. In our example scenario the user right clicks on the “Member Search” tool and then selects the “Change Name” option and provides the “New Screen Term” as the new name, as shown in Figure 4.33.

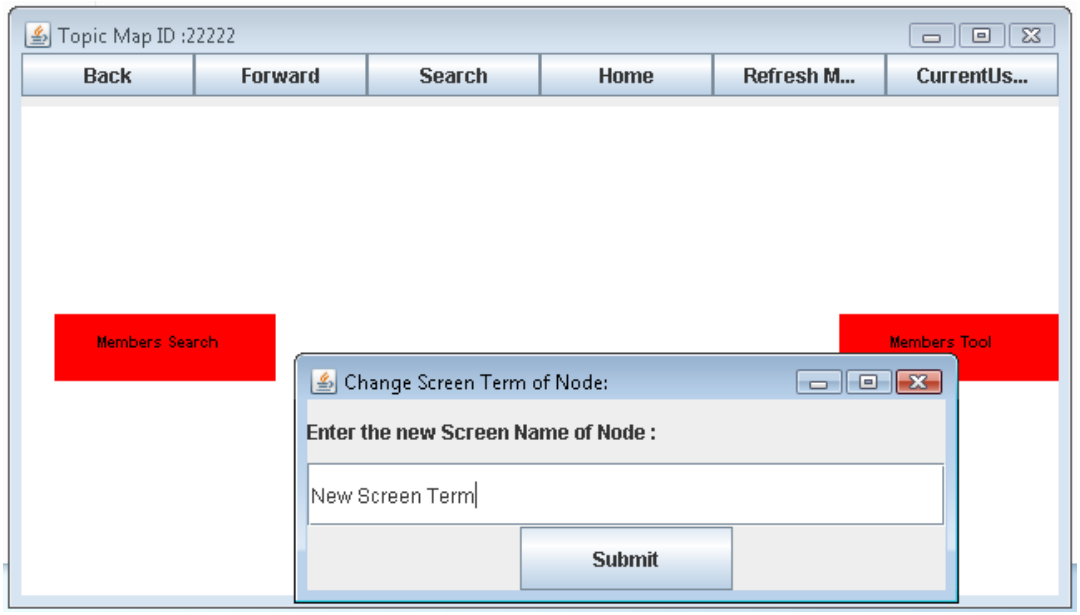

Figure 4.33User provides new Screen Name.

When the user clicks the Refresh, the name of the tool is changed to “New Screen Term” as shown in Figure 4.34. Now when the user uses the Information Retrieval tool, the “New Screen Term” would be used corresponding to this node.

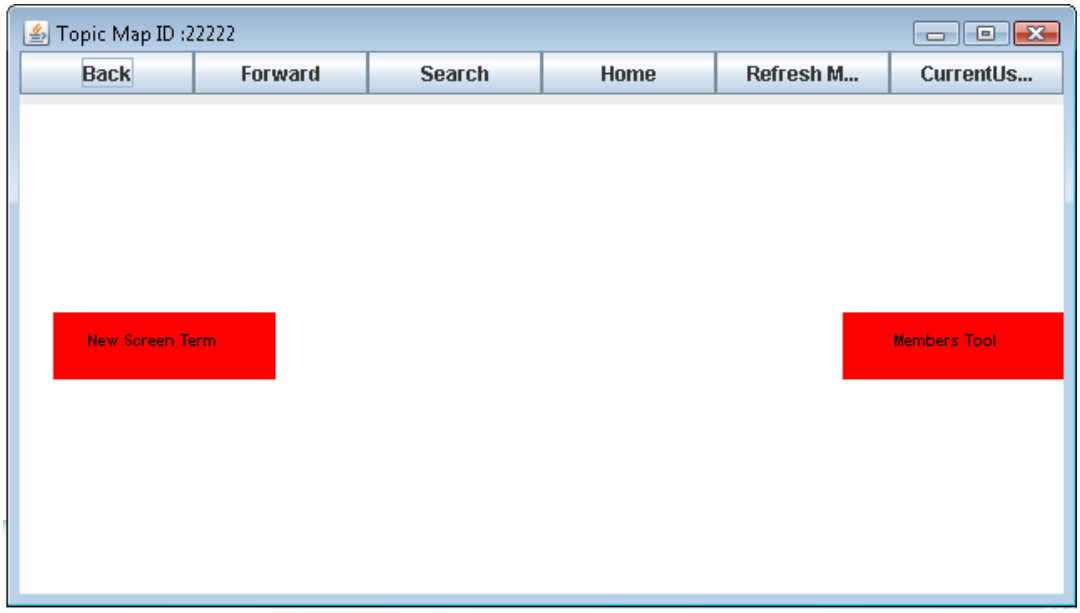

Figure 4.34 Screen name changed to “New Screen Term”.

When the user selects the Change the Path Label, then the user is provided with the old pathlabel "Search By Topic, Members, New Screen Term" as shown in the Figure 4.35. The user can modify this old pathlabel according to his requirement.

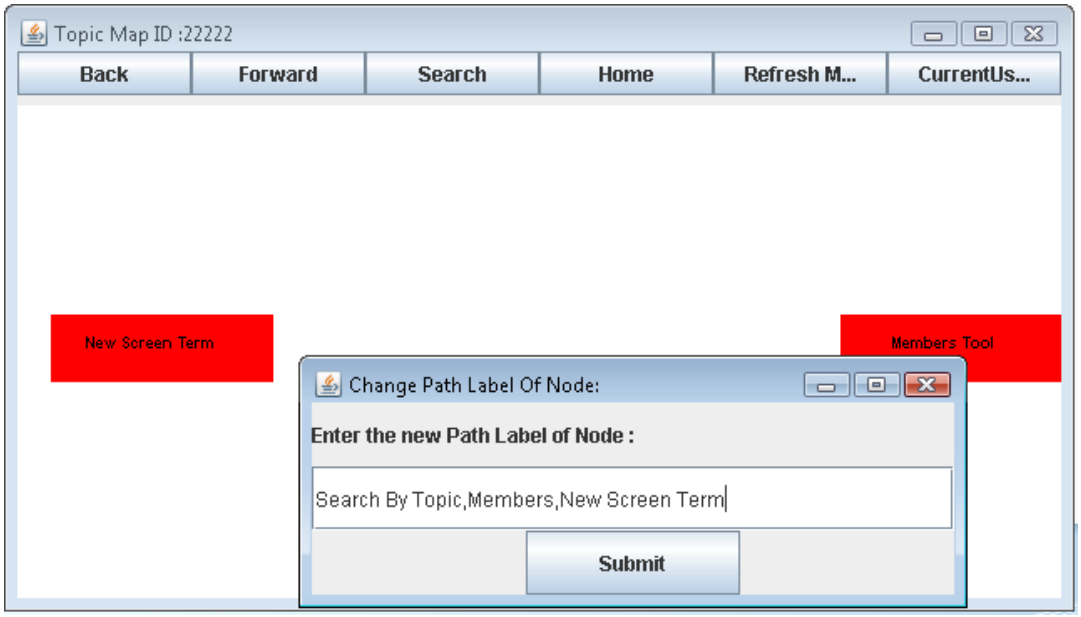

Figure 4.35 User can modify the old path label.

In our current example scenario, we modify the old pathlabel by removing the "New Screen Term" and then clicking on the Submit button as shown in the Figure 4.36. When the user clicks the Refresh, the changes are preserved.

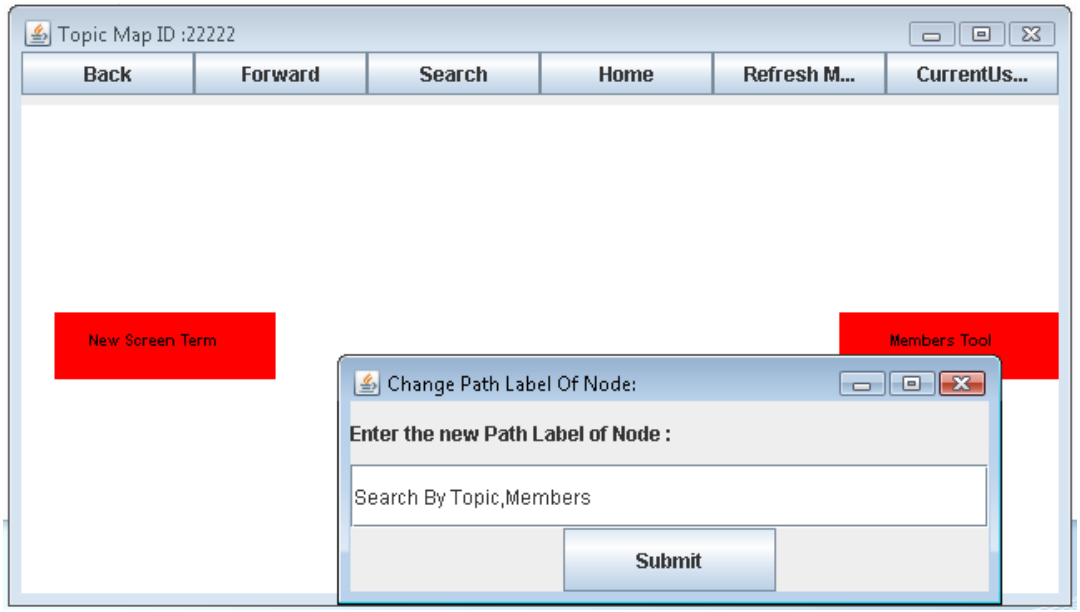

Figure 4.36 User modifies the path label.

4.7.7 MoveANode

The user right clicks on the “Email” topic node. Then the user selects the MoveANode operation from the menu. As shown in Figure 4.37.

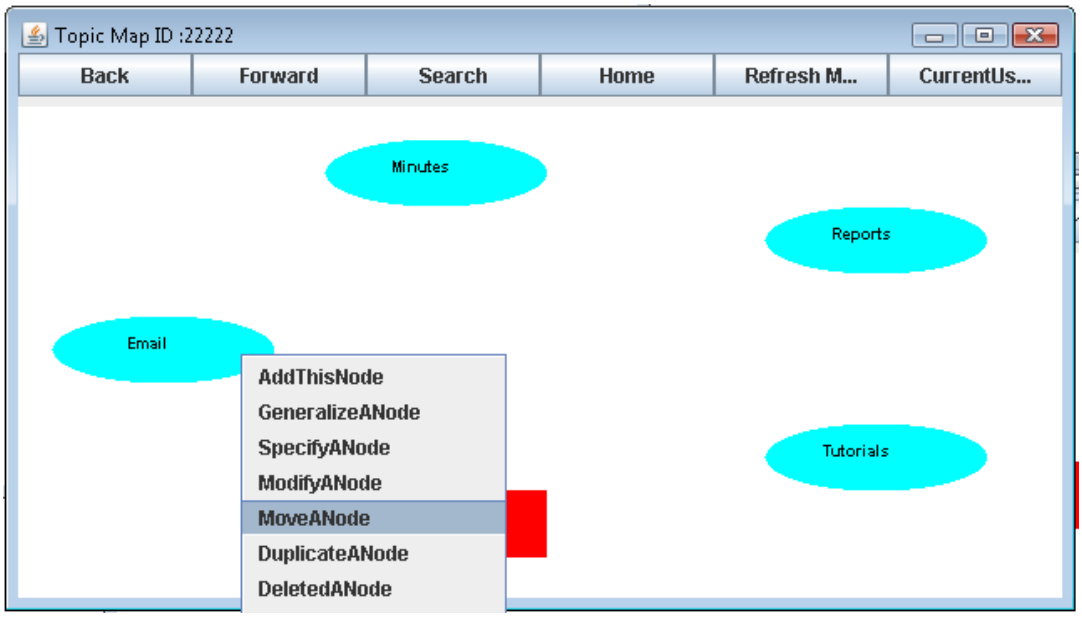

Figure 4.37 User selects the MoveANode operation.

Then the user navigates to the screen where he wants to move this node, using Forward, Backward Button or by clicking on the Nodes to reach appropriate screen. Then, the user right clicks on the free area on the screen where no node is present. The user selects the “CopyTheNode” operation from the menu as shown in the Figure 4.38

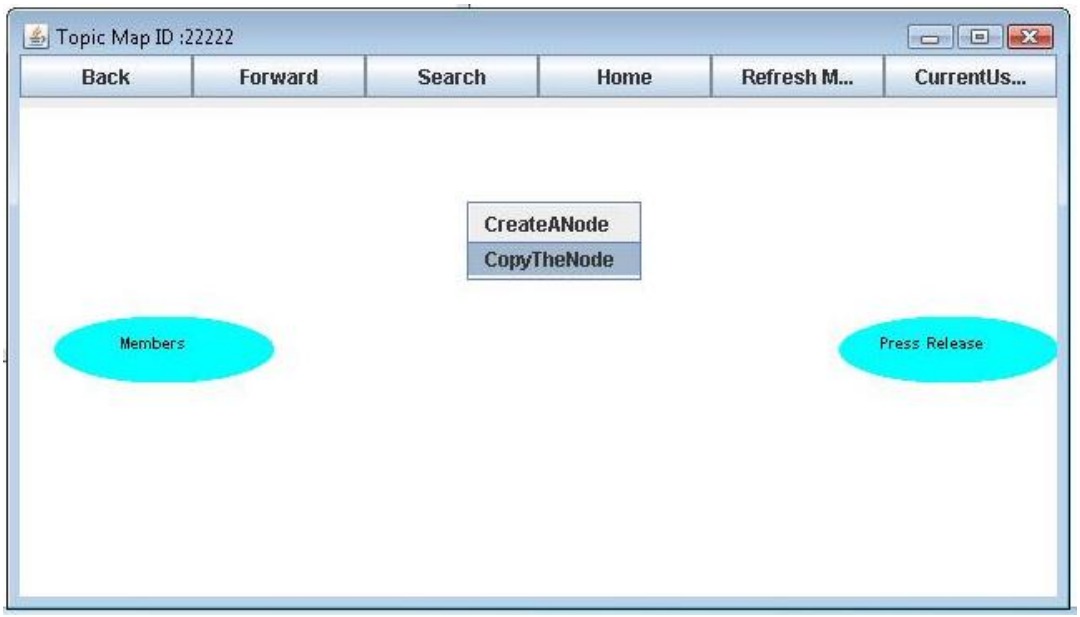

Figure 4.38 User selects the CopyTheNode operation.

After that the user clicks the refresh, the “Email” node is deleted from the old screen location and is moved to the new screen location as shown in the Figure 4.39.

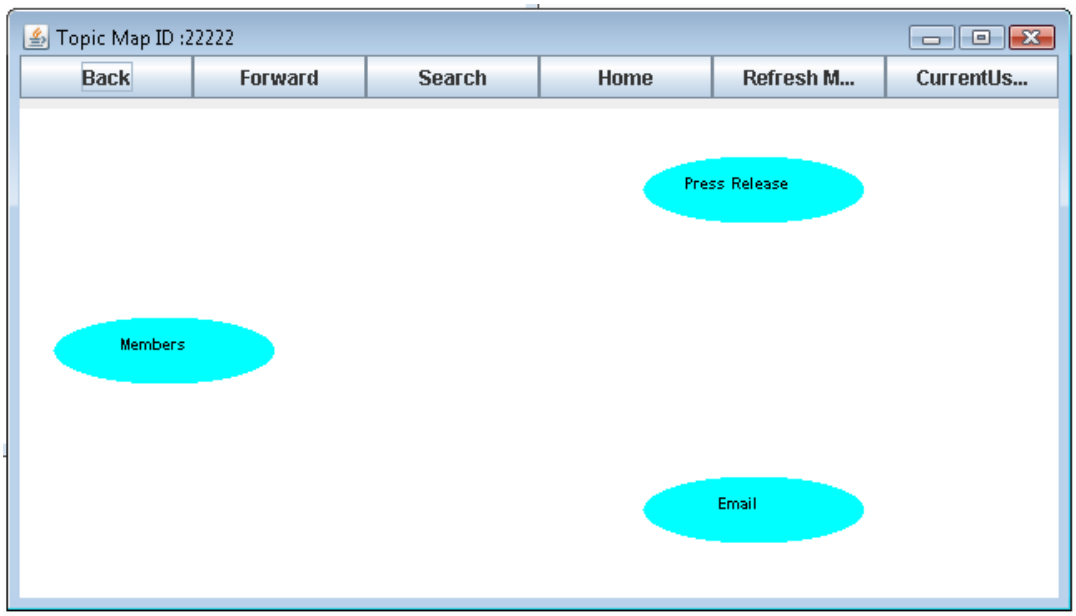

Figure 4.39 “Email” node copied to new screen location.

4.7.8 DuplicateANode

The user right clicks on the “Member Search” node. Then the user selects the “DuplicateANode” operation from the menu, as shown in Figure 4.40.

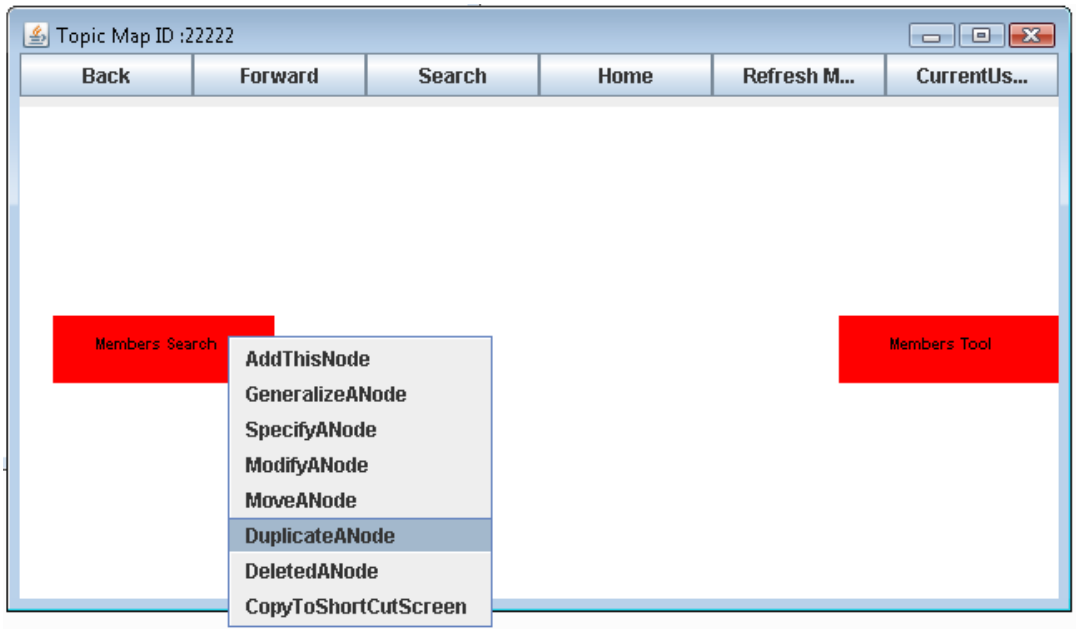

Figure 4.40 User selects the DuplicateANode operation.

Then the user navigates to the screen where he wants to duplicate this node, using Forward, Backward Button or by clicking on the Node to reach appropriate screen. Then, the user right clicks on the free area on the screen where no node is present. The user selects the CopyTheNode operation from the popup menu as shown in Figure 4.41.

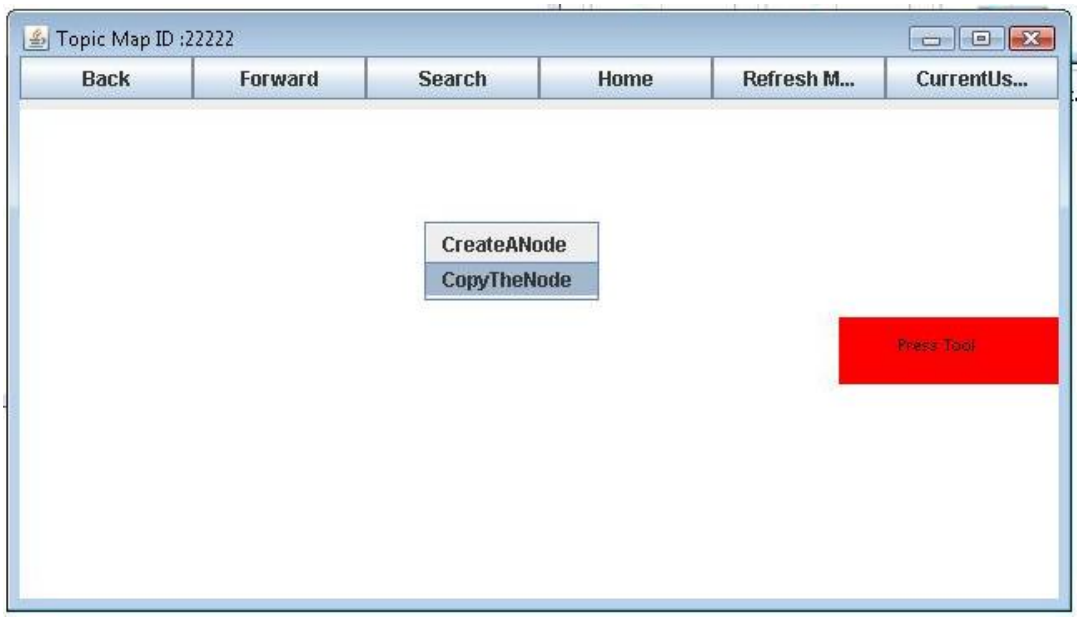

Figure 4.41 User selects the CopyANode operation.

When the user clicks the Refresh, the node is duplicated on the selected screen as shown in the Figure 4.42.

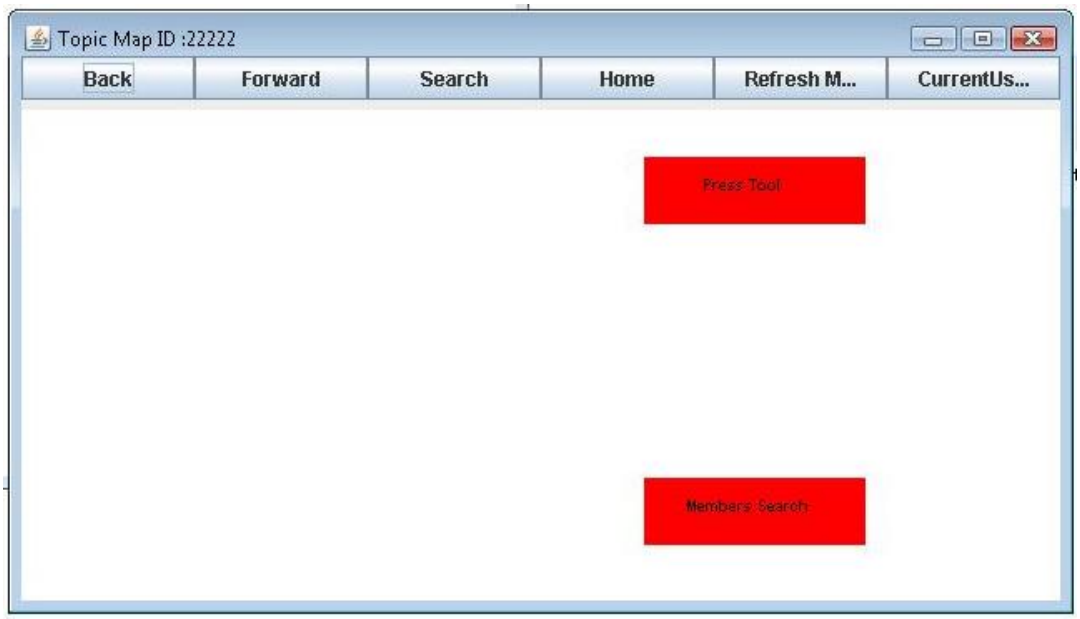

Figure 4.42 “Members Search” tool copied to new screen.

4.7.9 DeleteANode

The user right clicks on the node “Press Search”. Then the user selects the “DeleteANode” operation from the menu, as shown in the Figure 4.43.

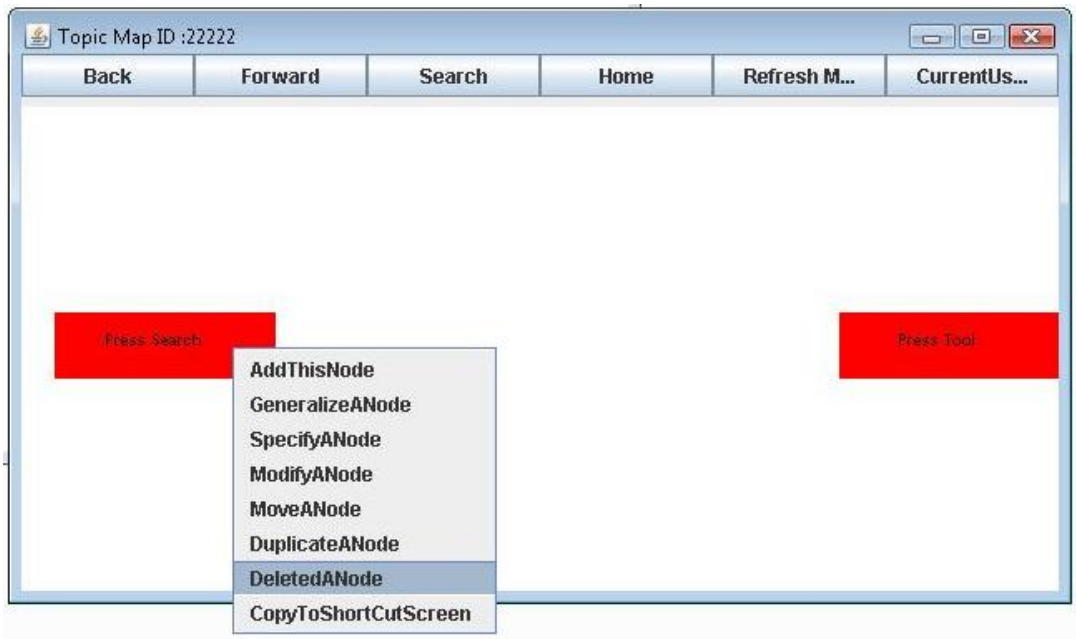

Figure 4.43 User selects the operation DeleteANode.

When the user clicks the Refresh, the node “Press Search” is deleted from the screen as shown in Figure 4.44.

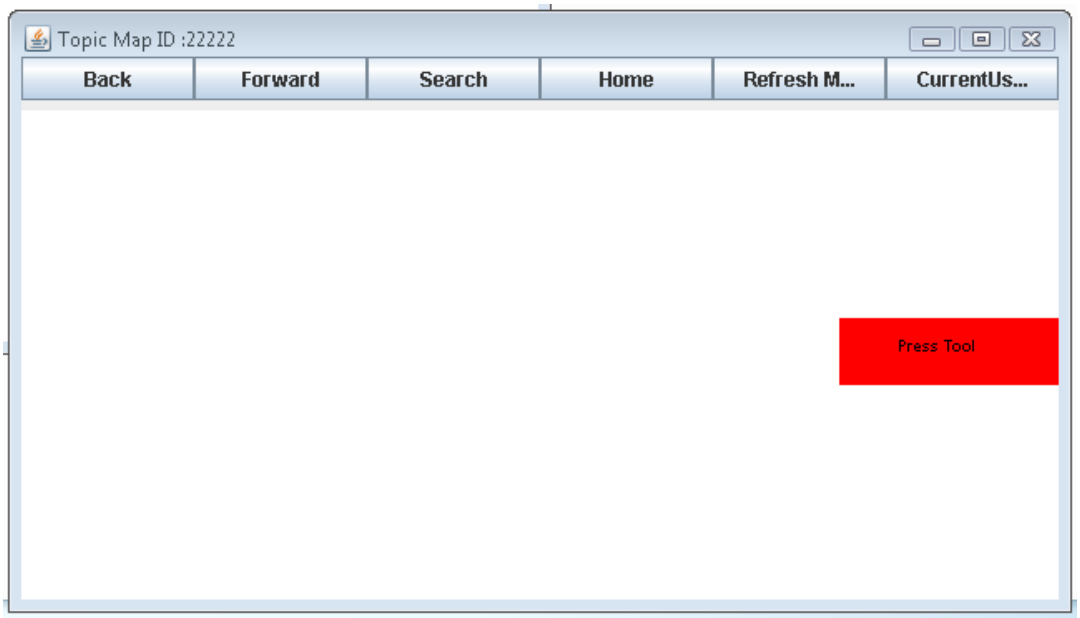

Figure 4.44 “Press Search” node deleted.

4.7.10 CopyToShortCutScreen

The user right clicks on the node “Cluster By Content” which he wants to copy on the Short Cut Screen. Then the user selects the “CopyToShortCutScreen” operation from the menu as shown in Figure 4.45.

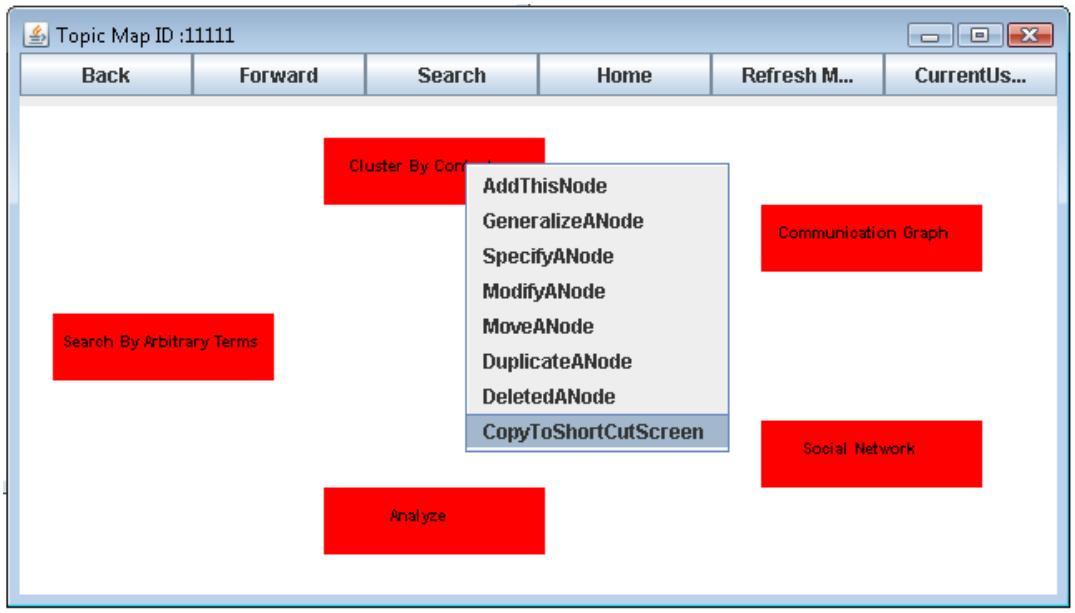

Figure 4.45User selects the CopyToShortCutScreen operation.

When the user clicks the Refresh, the node “Cluster By Content” is copied to the short cut screen and the changes are preserved, as shown in the Figure 4.47.

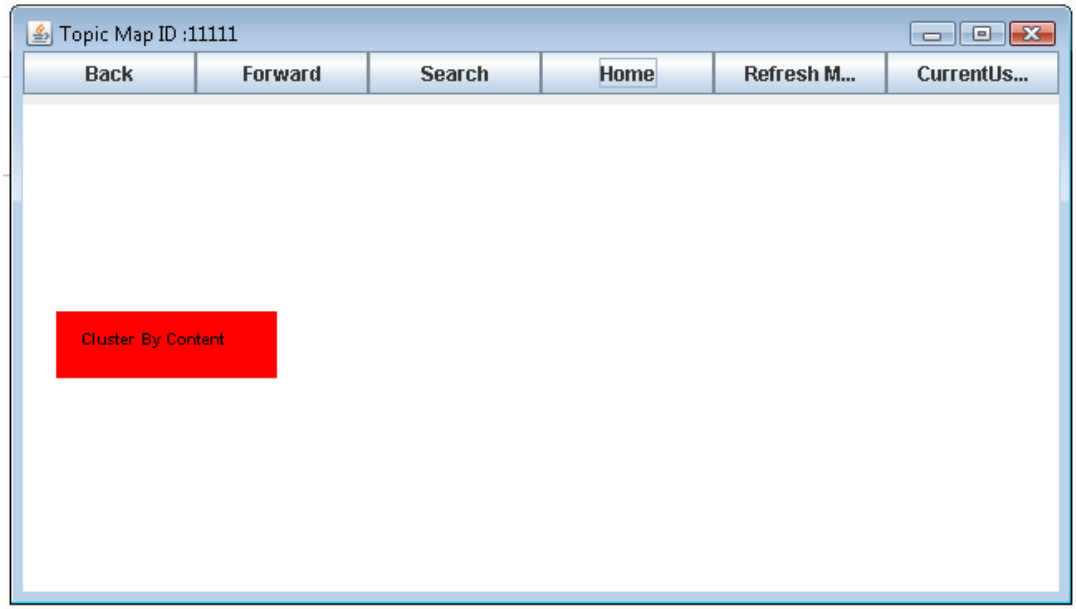

Figure 4.47 “Cluster By Content” node copied to ShortCut Screen.

In this chapter we discussed the implementation details, technologies and an example scenario in detail. In the next chapter we would discuss the conclusion and future scope of our project.

CHAPTER 5. CONCLUSION AND FUTURE WORK

In today's competitive environment, organizations strive to perform best. For this, they need to make optimum utilization of available resources. In this process, they are challenged to deal with an ever increasing and interconnected web of knowledge and information. This requires the use of proficient and effective tools to abstract the underlying knowledge. This knowledge may be extracted from collaborative tasks which may span across various groups within an organization and from myriad sources. Moreover, for the competitive advantage, building capabilities to access, conceptualize and reuse this knowledge is also important. We presented a topic map based model to support these requirements. The topic map based approach helps in identifying and mapping intellectual assets within the organization. It helps in generating new knowledge from resources within the organization, in making vast amounts of corporate information accessible in efficient manner and in sharing of best practices. Our prototype implementation showed how topic maps can be used at process levels and group levels. At group levels, the topic map supports dynamic changes to the structure through collaborative search and modification capabilities.

Currently a user can perform all the operations on the Group Topic Maps he/she has access to. In future, constraints can be put on the kind of operation an individual user can perform. In future models, the changes made by a particular user may be recorded and displayed when required. Moreover, recursive creation of new nodes may be implemented in future models. In addition, future prototypes may further expand the implementation of the Group Topic Map operations, for example it would be useful to provide users with screen level operations. The number of Tools and their implementation can be further expanded. Finally, development of a comprehensive prototype will allow us to test and analyze the model in an industrial or agency setting.

REFERENCES

[1] Argote, L., B. McEvily, et al. (2003). "Managing Knowledge in Organizations: An Integrative Framework and Review of Emerging Themes." *Management Science* **49**(4): 571-583.

[2] Choy, K. L., W. B. Lee, et al. (2005). "A knowledge-based supplier intelligence retrieval system for outsource manufacturing." *Knowledge-Based Systems* **18**(1): 1-17.

[3] Lee, T., M. Chams, et al. (1999). "Information Integration with Attribution Support for Corporate Profiles." *Proceedings of the eighth international conference on Information and knowledge management*., Kansas City, Missouri, United States, ACM Press.

[4] Nonaka, I. and N. Konno (1998). "The Concept of "Ba": Building A Foundation for Knowledge Creation." *California Management Review* **40**(3): 40-54.

[5] Miller, Les, Sree Nilakanta , Yunan Song , Lei Zhu , Ming Hua (2007), " Knowledge Management: Using Topic Maps in Organizational Memory.", *Proceedings of the Fourtieth Annual Hawaii International Conference on System Sciences* (CD-ROM), January 4-7, 2007, Computer Society Press, 2007, p204b (9 pages).

[6] Organizational Memory, http://en.wikipedia.org/wiki/Organizational_memory

[7] Walsh, J. P. and G. R. Ungson (1991). "Organizational Memory." *The Academy of Management Review* **16**(1): 57.

[8] Akgun, A. E., J. C. Byrne, et al. (2006)."Transactive memory system in new product development teams." *IEEE Transactions on Engineering Management* **53**(1): 95-111.

[9] Brockman, B. K. and R. M. Morgan (2003). "The role of existing knowledge in new product innovativeness and performance." *Decision Sciences* **34**(2): 385-419.

[10] Jennex, M. and L. Olfman (2002). "Organizational memory/knowledge effects on productivity, a longitudinal study." *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002.

[11] Ji, Y. G. and G. Salvendy (2004). "Interface methods for using intranet portal organizational memory information system." *Ergonomics* **47**(15): 1585-1597.

[12] Lesser, E. L. and J. Storck (2001). "Communities of practice and organizational performance." *IBM Systems Journal* **40**(4): 831-841.

[13] Stein E.W (2005). "Organizational Memory : Review of Concepts and Recommendations for Management. " *International Journal of Information Management* **15**(2), 17-32.

[14] Duncan ,R and Weiss , A (1979) "Organizational learning: implications for organizational design" *in STAW, BM(ED) Research in Organizational Behaviour JAI Press*, Greenwich, CT, pp 75-123.

[15] Prahalad , CK and Hamel, G(1990) "The core competence of the corporation." *Harvard Business Review*, May-June , 79-91.

[16] Christopher A. Bartlett and Sumantra Ghoshal (2002) "Building Competitive Advantage Through People" *MIT Sloan Management Review Jan-Feb.*

[17] Ackerman, M. and C. Halverson (2004). "Organizational Memory as Objects, Processes, and Trajectories: An Examination of Organizational Memory in Use." *Computer Supported Cooperative Work (CSCW)* **13**(2): 155-189.

[18] Hutchins, E. (1995). *Cognition in the Wild.* MIT Press.

[19] Hollan, J., E. Hutchins, et al. (2000). "Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research." *ACM Transactions on Computer-Human Interaction (TOCHI)* **7**(2): 174-196.

[20] Wenger, E. (1998). "Communities of Practice: Learning as a social system." *The Systems Thinker* **9**(5).

[21] Smolnik, S. and I. Erdmann (2003). "Visual navigation of distributed knowledge structures in groupware-based organizational memories." *Business Process Management Journal* **9**(3): 261.

[22] ISO standard ISO/IEC 13250 Topic Maps http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf

[23] Rath, Dr. Hans Holger and Pepper, Steve (1999), "Topic Maps: Introduction and Allegro"; *Proceedings of the Markup Technologies* 99, Philadelphia, USA.

[24] Steiner, K., W. Essmayr, et al. (2001). "Topic Maps - an Enabling Technology for Knowledge Management.", *12th International Workshop on Database and Expert Systems Applications*, 2001.

[25] Ramalho, J.C., Librelotto, G.R., and Henriques, P.R. (2006), "Metamorphosis - A topic maps based environment to handle heterogeneous information resources", *First International Workshop on Topic Map Researchand Applications*, TMRA 2005, Springer, 2006, pp. 14-25.

[26] Tsampoulatidis, I., Tzovaras, D., and Strintzis, M.G. (2004), "Ontology-based E-government thematic services based on topic maps", *On the Move to Meaningful Internet Systems 2004: Otm 2004 Workshops, Proceedings*, 2004, pp. 569-580.

[27] Korthaus, A., S. Henke, et al. (2006). "A Distributed Topic Map Architecture for Enterprise Knowledge Management", *5th IEEE/ACIS Conference on International Computer and Information Science*, ICIS-COMSAR 2006.

[28] Hibernate, https://www.hibernate.org/

[29] Soap, http://www.w3.org/TR/2001/WD-soap12-20010709/

[30] Axis2, http://ws.apache.org/axis2/

[31] WSDL, http://www.w3.org/TR/wsdl

APPENDIX

Figure A.1 shows Partial content of the OMTerms.txt. The *OMTerms.txt* file is a text file, it stores all the initial information required to setup and creates the Topic Maps. The given content has data for the topic map "22222". The first time the Server is started it reads the text file *OMTerms.txt*, its *GenerateFile* class reads and parses this text file.

```
22222:01,001-Search Memory Collections:topic,002-Search By Topic:topic,003-Find
Experts:tool[ExpertTool],004-Search By Arbitrary Terms:tool[IRTool],005-Tutorial Search:tool[IRTool],006-
Committee Search:tool[IRTool],999-Short Cut:topic
22222:0101,007-Email:topic,008-Minutes:topic,009-Reports:topic,010-Tutorials:topic,004-Search By Arbitrary
Terms:tool[IRTool]
22222:010101,054-Search By Arbitrary Terms:tool[IRTool],055-Cluster By Content:tool[ClusterTool],056-
Communication Graph:tool[GraphTool],057-Social Network:tool[SocialNetworkTool],058-
Analyze:tool[AnalysisTool]
22222:010102,004-Search Term:tool[IRTool],013-Cluster By Term:tool[ClusterTool],014-Communication
Graph Term:tool[GraphTool],015-Social Network Term:tool[SocialNetworkTool],016-Analyze
Term:tool[AnalysisTool]
22222:010103,074-Search Content:tool[IRTool],073-Cluster By Content:tool[ClusterTool],075-Communication
Graph:tool[GraphTool],076-Social Network Content:tool[SocialNetworkTool],077-Analyze
Content:tool[AnalysisTool]
22222:010104,017-Members:topic,018-Press Release:topic
22222:01010401,060-Members Search:tool[IRTool],061-Members Tool:tool[IRTool]
22222:01010402,062-Press Search:tool[IRTool],063-Press Tool:tool[IRTool]
22222:0102,017-Members:topic,018-Press Release:topic,019-Transcripts:topic,020-Annual Report:topic,049-
Minutes:topic,021-Expand Topic:tool[ExpandTopicTool]
22222:010201,060-Members Search:tool[IRTool],061-Members Tool:tool[IRTool]
22222:010202,062-Press Search:tool[IRTool],063-Press Tool:tool[IRTool]
22222:010203,064-Transcripts tool1:tool[IRTool],065-Transcripts Tool2:tool[IRTool]
22222:010204,066-Annual tool1:tool[IRTool],067-Annual Tool2:tool[IRTool]
22222:010205,022-General Info:topic,023-Consumer Related:topic,024-Business Related:topic,025-
Government Related:topic,026-Labor Demand:topic,030-Expand Topic:tool[ExpandTopicTool],042-
Payroll:topic,068-Retrive:topic
22222:01020501,032-Meeting Time:tool[IRTool],033-People Present:tool[ListPeopleTool],034-
Agenda:tool[IRTool]
22222:01020502,035-Real Personal Consumption Expenditure:tool[IRTool],036-Consumer Price
Inflation:tool[IRTool]
22222:01020503,037-Business Expenditure:tool[IRTool],038-Industrial Production:tool[IRTool],039-Housing
Market Activity:tool[IRTool],040-Trade Deficit:tool[IRTool]
22222:01020504,041-Government Spending:tool[IRTool]
22222:01020505,042-Payroll:topic,043-Unemployeement Rate:tool[IRTool]
22222:01020507,047-Term Search:tool[IRTool],048-Retrive Document:tool[IRTool]
22222:01020508,088-Term:topic(12;13;14)
22222:0102050501,067-Payroll Tool:topic(7;8;9)
```

Figure A.1Partial Content of OMTerms.txt

The figures A.2 and A.3 shown below provide information about the packages and the classes present in our system. Figure A.2 gives details of the packages and classes present on the client side. Figure A.3 give details of the packages and classes present on the server side.

Package	Classes
dataTypes	Node Screen ShortCutScreen ToolListElement TopicListElement TopicMapDataMaps TopicMapDataMapsFactory
dynamicTopicMap	CurrentScreenUtilities CurrentUserScreen DataForCopyNode FireNode GenerateTopicMap GraphicsPanelFactory SetupScreen StartUpTopicMap UserStart WindowUtilities
databaseAccess	GetTopicMapList TmClient
operations	ChangeName ChangePathLabel ChangeResourceTool CreateNode CreateNodeUtilities CreateTool CreateTopic GeneralizeANode ModifyANode SpecifyANode

Figure A.2 Client side packages and classes.

Package	Classes
databaseAccess	AccessStoredData AccessTopicMapStoredData GenerateFile HibernateSessionFactory InterfaceAccessTopicMapStoredData ProcessPathLabel
dataTypes	CurrentUsers CurrentVersion GenerateNodeNo Node Screen ShortCutScreen TopicMapEditDetails TopicNodeId TopicResourceListElement ToolListElement User UserDetails

Figure A.3 Server side packages and classes.

ACKNOWLEDEGEMENT

I am profoundly grateful to my major professor Dr. Les Miller for his supervision, support and encouragement. His guidance was integral and helpful throughout my studies at the Iowa State University.

I would also like to thank Dr. Gadia and Dr Nilakanta for their willingness to serve on my POS committee and for their invaluable comments and guidance.

Last but certainly not least, I express my deep gratitude to my parents, for their sacrifices, love and support.

ALOK SHARMA