INIINTERACTIVE ANIMATED CONCEPT MAP GENERATION

- USING DYNAMIC ANIMATIONS

By

SUNEEL VANA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment of the Requirements for the

Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2005

ACKNOWLEDGMENTS

I would like to express my sincere appreciation and gratitude to all the people who may be related to this research directly and indirectly. Special gratitude should be given to my advisor Dr. Y. Alp Aslandogan, for his excellent advice and support in this research. His tireless patience and great understanding of visualization are the keys for the success of this thesis. This thesis work could not have been possible without his constant guidance, patience and motivation. Throughout my thesis-writing period, he provided encouragement, sound advice, good teaching, and lots of good ideas.

This acknowledgement is incomplete without mentioning my team members and lab members who helped me throughout the work. Among them, Mitali Gokhale, Gauri Mahajani, Pallavi Boppana and Emere Celebi have been particularly helpful and generous with their time and expertise.

I am grateful to all my friends for their encouragement and the moral support. I wish to thank my family members for providing immense support. Finally, I am forever indebted to my parents for their understanding, endless patience and encouragement when it was most required. I dedicate this thesis to my family.

November 11, 2005

ABSTRACT

IINTERACTIVE ANIMATED CONCEPT MAP GENERATION

- USING DYNAMIC ANIMATIONS

Publication No. ______

Suneel Vana, M.S.
The University of Texas at Arlington, 2005

Supervising Professor: Alp Aslandogan

We present visualization too, which is useful for the illustration of educational concepts using interactive animated concept maps. This System serves as a presentation tool which provides a collection of multimedia elements (images, animations, audios clips and video clips), a collection of animated presentation templates and an interface by using which users can retrieve the multimedia elements from the multimedia database and generate the dynamic animations by placing the selected elements into the placeholders in the templates. This system generates the animated presentation by replacing embedded multimedia objects and their properties as desired by its users. Content objects are obtained from a multimedia database that is populated by a web search agent which conducts concept based searches. A domain knowledge base provides the base queries for automatic search initiation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

The purpose of this thesis is to build a visualization-oriented presentation system for illustrating educational concepts. This system serves as a presentation tool which provides a collection of multimedia elements (images, animations and videos), a collection of animated presentation templates and an interface by using which users can retrieve the multimedia elements from the database and generate the dynamic animations by placing the selected elements into the placeholders of the templates.

The system aims at generating multimedia presentations such as graphical animations using presentation templates by replacing embedded multimedia objects and their properties as desired by the users. This helps the educator in better understanding of the concepts. A multimedia information extraction agent automatically populates the database with content objects such as images, audio clips, video clips, graphic objects and other animations by means of a novel approach named “concept expansion”. A domain knowledge base provides the base queries for automatic search initiation. We employ both content and context analysis to index multimedia elements found in large collections such as the World Wide Web.

The main features and goals of the system are discussed below. Section 3.3 talks about the presentation templates. Section 3.4 gives general information about the database design and how the multimedia elements are stored in the database. Section 3.5 gives the detailed description of the system design and architecture. Section 4 discusses about the implementation of the complete system, chapter 5 discusses about the related work and chapter 6 discusses about the conclusion and future work.

1.1 Visualization

Before the era of computers started, the term visualization meant constructing a visual image in the mind. But now it has come to mean something more like a graphical representation of data and concepts. Visualization is the process of transforming data, information, and knowledge into visual form making use of human's natural visual capabilities [1]. Visualization is the process of creating and manipulating graphical representations of data or information. Visual representations help in better understanding of the problem we are studying or the concept we are learning also it helps in extracting the hidden patterns in information. It has been said throughout time that a picture could be worth a thousand words. Picture can convey an overall message much better than an illustration list. A rich visualization will facilitate the viewer with rapid understanding of the concept and more importantly, an understanding of the overall phenomenon behind the concept.

One of the greatest advantages of visualization is the sheer quantity of information that can be rapidly interpreted if it is presented well. Visualization provides the ability to comprehend huge amounts of data. Visualization allows the perception of emergent properties that were not anticipated. The need for information visualization is clearly explained in [2]. Visualization reveals things about data as well as how the data is collected. Visualization plays a very important role in quality control, if the data is visualized in a proper way it elevates the errors automatically. Visualization helps in perception of patterns and also helps in relating them to the local features. Visualization helps in analyzing both large-scale and small-scale features of the data. Simply saying visualization helps in observing the patterns in data, exploring the conceptual information and also in explaining the concepts.

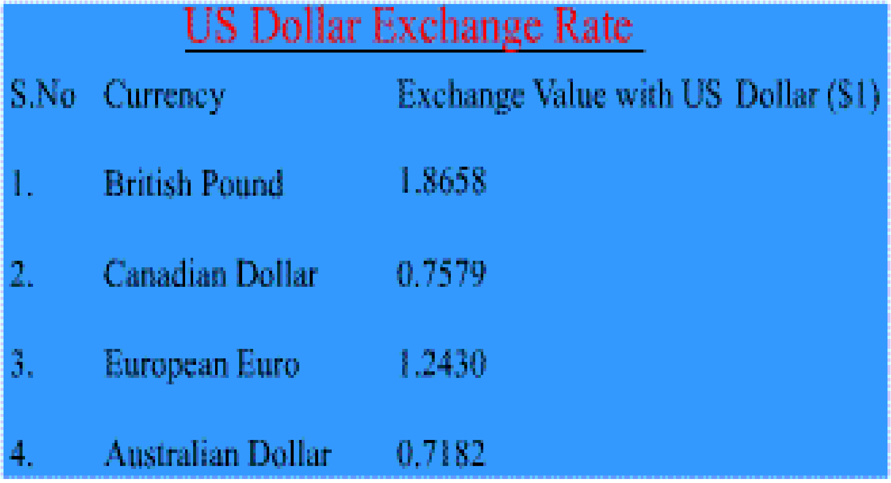
Classification of visualization techniques is often based on the type of the data that has to be visualized and type of the user tasks that the visualization supports. Visualization techniques can also be divided into surface rendering techniques, and (direct) volume rendering techniques [3]. Based on the dimension of the domain of the quantity that is visualized, visualization techniques are classified as

1. Direct viewing of data
2. 2-Dimensional visualization
3. 3-Dimensional Visualization
4. Multi-dimensional Visualization.

1.1.1 Direct Viewing of Data

Direct viewing of data is the process where we do not use any visualization technique to represent the data. This is the basic way of representing the data as it appears. This is also known as 1-D visualization. It is the way of representing the raw data as it was collected. A collection of data related to currency exchange conversion rates between different currencies is an example of direct representation of data.

Figure 1.1 shows the US Dollar exchange rate with the currency of Britain Pound, Canadian Dollar, European Euro and Australian Dollar. From this way of representation we cannot interpret good amount of information. It is also difficult to find out the relation between all parameters by looking at this data. There is lots of hidden information, which we cannot comprehend from this representation.



US Dollar Exchange Rate

S.No	Currency	Exchange Value with US Dollar ($1)
1.	British Pound	1.8658
2.	Canadian Dollar	0.7579
3.	European Euro	1.2430
4.	Australian Dollar	0.7182

Figure 1.1 Visualization of Quantitative Information

1.1.2 2-D Visualization

The 2-dimensional information visualization environment combines 2-dimensional data representation and two dimensional data types. This technique uses 2-dimensional data types to represent data in 2-dimensional plane. Direction, location, size, and distance are features that can be effectively illustrated using a 2-D visualization environment [3]; these features are the dimensions, which we represent in a 2-dimensioanl plane. One most important thing in 2-D visualization is that in the two attributes that we are representing one is always fixed and the other attribute value will keep changing. The best way to define 2-dimensional data is to identify what are the two dimensions of the data that we are focusing on and then representing those two attributes in a 2-D plane. 2-D representation mainly helps in comparing the same type of data associated with two objects. For example if we want to know the population growth year 2003 to year 2004, we can represent the year attribute on one dimension and the population attribute on the other dimension and can draw a scatter plot. In this scenario the year is attribute is fixed and the population attribute will be varying.

If we take the example described in section 1.1, we can represent the same tabular data using a bar chart on a 2-dimensional plane. In this scenario there are two dimensions one is the currency name and the second one is its exchange rate with US dollar. Currency name is a fixed variable where as its exchange rate will be changing from time to time. Every time when the values change we draw a new chart. The 2-

dimensional representation of the data will be as show in Figure 1.2. In Figure 1.2 X-axis of the plane represents the currency name and Y-axis on the plane represents the exchange rate with US Dollar. By using this representation we can comprehend lots of information that was hidden in the 1-D representation.

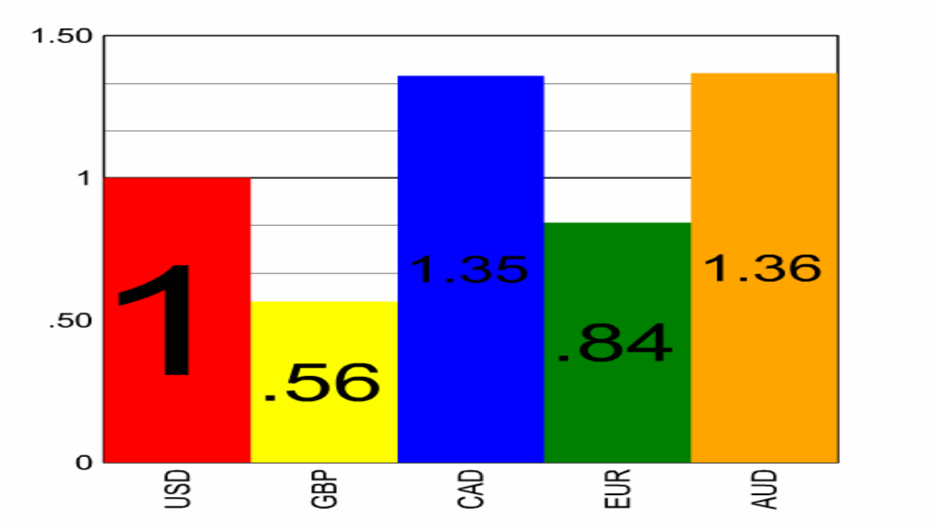


Figure 1.2 2-D Visualization of Quantitative Information

If we take the example described in section 1.1, we can represent the same tabular data using a bar chart on a 2-dimensional plane. In this scenario there are two dimensions one is the currency name and the second one is its exchange rate with US dollar. Currency name is a fixed variable where as its exchange rate will be changing from time to time. Every time when the values change we draw a new chart. The 2-dimensional representation of the data will be as show in Figure 1.2. In Figure 1.2 X-axis of the plane represents the currency name and Y-axis on the plane represents the exchange rate with US Dollar. By using this representation we can comprehend lots of information that was hidden in the 1-D representation.

1.1.3 3-D Visualization

3-D visualization is a term used to describe viewing real world objects such as the human body, buildings, or molecules for the purpose of extracting information, or to represent quantitative information (Figure 1.3) in 3-Dimensional plane. 3-D visualization helps in representing three-dimensional (3-D) information in a two-dimensional (2-D) space (e.g., the display screen). Whereas pixel location in a 2-D graphic is defined only on the X-Y axes, a third dimensional pixel adds a third axis, called the depth property. This form of visualization should be applied when viewing the object in 3D or rather what is inside the object is key to the desired users' task. This type of visualization is also called as volume visualization [4].

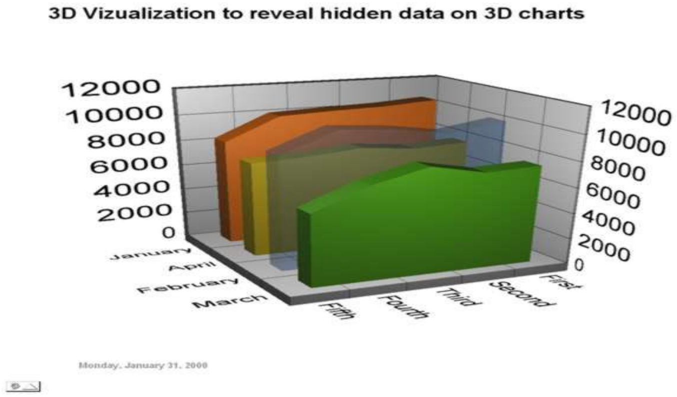


Figure 1.3 3-D Visualization of Quantitative Information

3-D visualization involves rendering of real world 3D objects into some form of computerized 3D representation, whether it be projected on the 2 dimensional computer screens or viewed through immerse virtual reality equipment. 3-D visualization is further classified into few sub categories. The most important category among these is the scientific visualization of real world 3D objects like those in volume, medical, and even architectural visualization. (Ex: Molecules, the human body and the interior of a building share complex relationships with other items). These are kind of objects that people may want to view the inner structures before physically entering. The ability to show the inner-structures of the object, without physically entering it places 3-D visualization way beyond the older techniques. This is the key functionality of 3-D visualization. 3D visualization provides the ability to navigate up, down, forward or backward once immersed into the given environment.

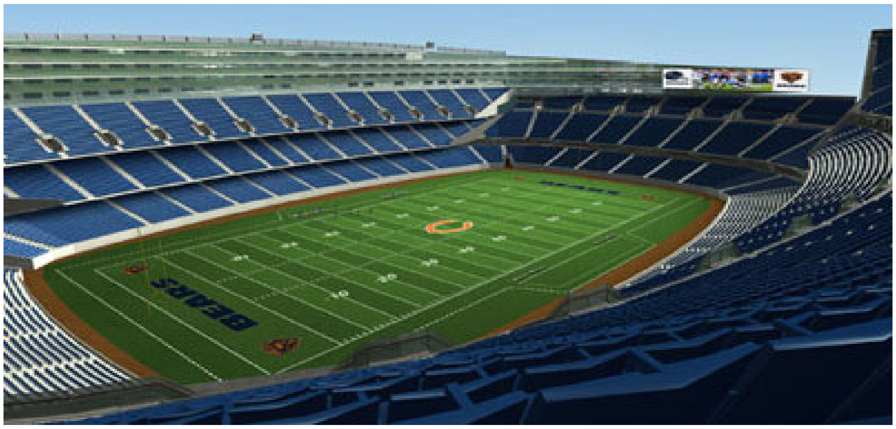


Figure 1.4 3-D Architectural Visualization

1.1.3 N-D Visualization

Multi-dimensional information visualizations present data that is not primarily spatial. The number of attributes of a given item in the collection is more than three. Visualizing multi-dimensional data has great impact on engineering, science and business decision-making. Example applications of multi-dimensional visualization schemes may use stock market statistics, factory production line data sets, a movie database, and almost any abstract and statistical information about any phenomenon [5].

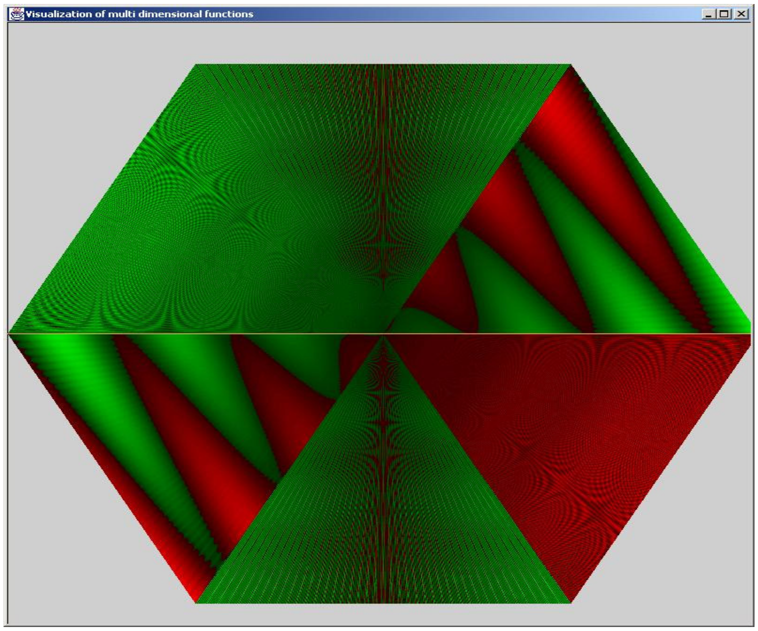


Figure1.5 N-D Visualization

Animation is also a technique in Multi-Dimensional visualization. Users can comprehend more information through an animation rather than still pictures. Attributes in multi-dimensional visualizations should have approximately equal weights. One, two, three dimensional and temporal information visualization schemes can be viewed as a subset of multi-dimensional information visualization. Scientific visualization also deals with multi-dimensional data but most of the data sets used in this field use the spatial attributes of the data for visualization purposes.

For example, Computer-Aided Tomography Systems, Computer-Aided Design Systems, and many of the Geographical Information Systems use either the Cartesian coordinate system or the geographical coordinates of the data to achieve a reasonable visualization of the data. The technique of using Star Coordinates [6] to visualize multi-dimensional data is a very good example of Multi-Dimensional visualization. Star coordinates arrange coordinates on a circle sharing the same origin at the center. It uses simply points to represent data treating each dimension uniformly at the cost of coarse representation. This technique provides valuable insight on several real data sets for cluster discovery and multi-factor analysis task.

1.2 Problem Statement

Today's educators face a dilemma: On the one hand they have a classroom full of multimedia-hungry students with short attention spans. To match this appetite they have access to enormous amounts of educational multimedia content available on the web and other sources. On the other hand, they have heavy teaching loads including test coaching and evaluation, which do not leave them sufficient time to master multimedia-authoring tools or to search for the content material. So they need a presentation system that is designed to empower educators in overcoming this dilemma [7]. The presentation system should also provide the mapping between the curriculum and multimedia content. To illustrate our approach imagine you as a science teacher at the junior high school level. You would like to illustrate the concept of “life cycle” for various animals. You would like to show similarities and differences among different types of animals. You might want to give an animated presentation that shows the stages of life cycles in sequence, possibly with repetition. Wouldn’t it be great to be able to open up a digital curriculum, point to the relevant topic, get a ready-made animated presentation, customize it for a few sample animals by dragging and dropping pictures of animals, and make these available on the web for a computer lab hour?

The motivation for building a visualization-oriented presentation system comes from interacting with educator and teachers at educational institutes. Teachers as well as students are interested in interactive visualization-oriented presentation system, which allow them to create different presentation for different concepts using animated templates and multimedia elements. This also allows the users to compare related concepts and explore relationships among the concepts. Our subjective experience is confirmed by interacting with teachers and students at few schools.

1.2 Thesis Organization

The presentation system is designed around a web/application server that acts as a mediator for six other components as illustrated in various sections. The remainder of this thesis is organized as follows. Section 2.1 gives an overview on vector and bitmap graphics and their differences. Sections 2.2 and 2.3 give the background information about Macromedia Flash, Jgenerator respectively. In sections 2.4 we describe the architecture of JDBC, and explain how the JDBC works with different RDBMS. Section 2.5 gives a brief introduction to MySQL and about the BLOB data type was given in these sections. Chapter 3 describes the design and architecture of the system in detail. In detail sections 3.1 and 3.2 discuss about Concept Map and Topic Map respectively. Section 3.3 explains about Presentation Templates where as section 3.4 explain about database design. Section 3.5 discusses about the system architecture and design. Chapter 4 explains about the implementation. Chapter 5 explains about the related work and chapter 6 explains about the conclusion and future work.

CHAPTER 2

BACKGROUND INFORMATION

This chapter explains about the background software applications that are used in this thesis. Section 2.1 gives an overview on vector and bitmap graphics and their differences. Section 2.2 gives a brief introduction to Macromedia Flash. Section 2.3 gives an overview about Jgenerator. Sections 2.4 and 2.5 describe about database management system JDBC and MySQL database respectively.

2.1 Vector and Bitmap Graphics

Computers display graphics in either vector or bitmap format. Understanding the difference between the two formats can help us work more efficiently. We can generate vector graphics using macromedia flash. In flash we can also import and manipulate vector and bitmap graphics that have been created in other applications. Vector graphics describe images using lines and curves, called vectors, which also include color and position properties [8]. For example, the image of a leaf is described by points (As shown in Figure 2.1) through which lines pass, creating the shape of the leaf's outline.

The color of the leaf is determined by the color of the outline and the color of the area enclosed by the outline. When we edit a vector graphic, we modify the properties of the lines and curves that describe its shape.

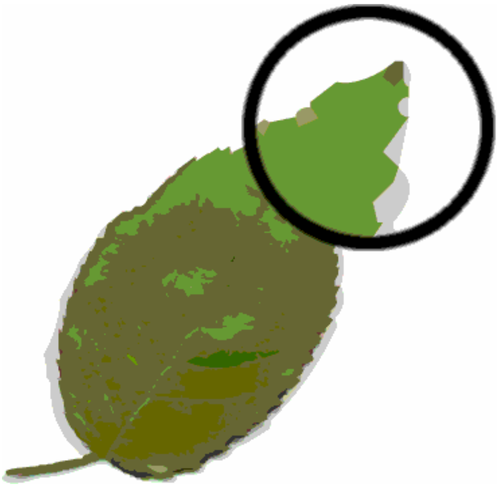


Figure 2.1 Vector Graphics

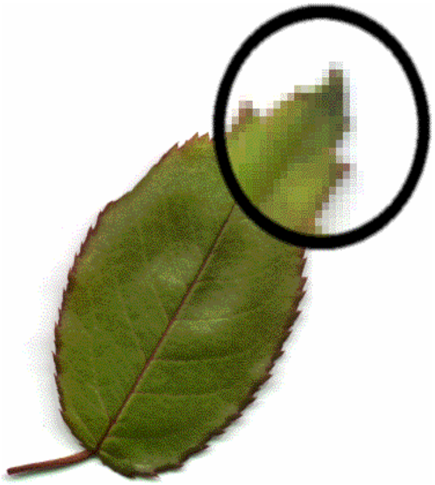


Figure 2.2 Bitmap Graphics

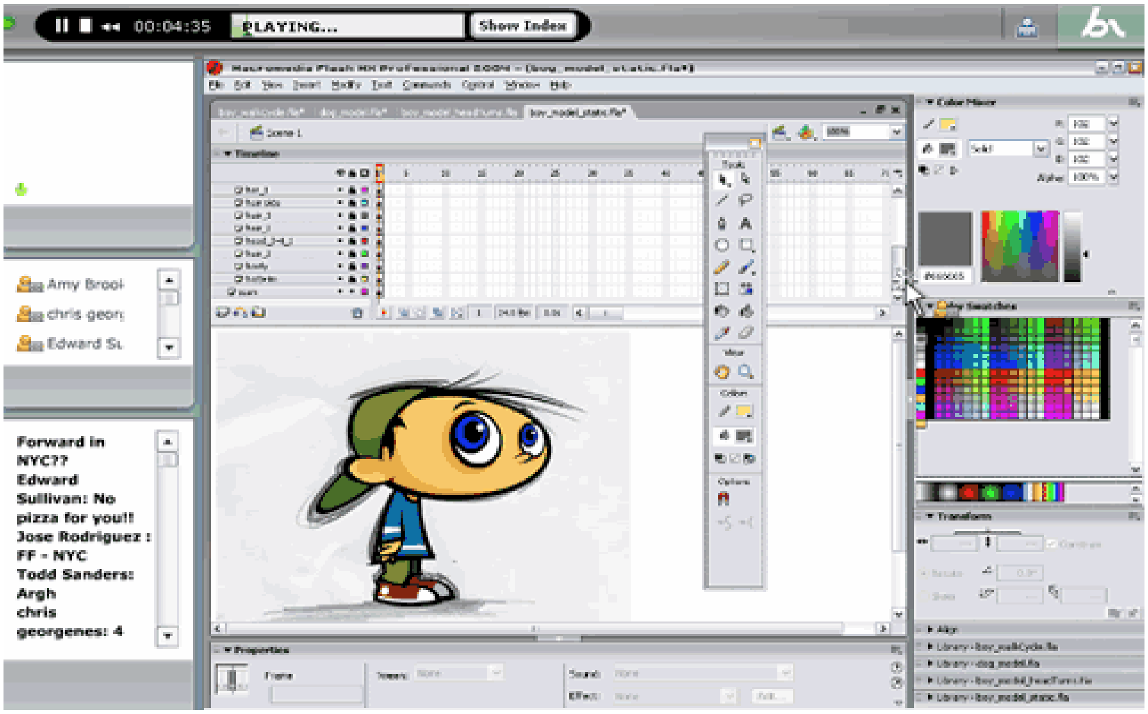
We can move, resize, reshape, and change the color of a vector graphic without changing the quality of its appearance. Vector graphics are resolution-independent, meaning they can be displayed on output devices of varying resolutions without losing any quality [9]. Bitmap graphics describe images using colored dots, called pixels, arranged within a grid. For example, the image of a leaf (As shown in Figure 2.2) is described by the specific location and color value of each pixel in the grid, creating an image much in the same manner as a mosaic. When we edit a bitmap graphic, we modify pixels, rather than lines and curves. Bitmap graphics are resolution-dependent

[4], because the data describing the image is fixed to a grid of a particular size. Editing a bitmap graphic can change the quality of its appearance. In particular, resizing a bitmap graphic can make the edges of the image ragged as pixels are redistributed within the grid. Displaying a bitmap graphic on an output device that has a lower resolution than the image itself also degrades the quality of its appearance.

2.2 Flash

Macromedia Flash is a multimedia graphics program especially to generate animations. Flash allows designers and developers to integrate video, text, audio, and graphics into immersive, rich experiences that deliver superior results for interactive marketing and presentations, E-learning, and application user interfaces. One of the strengths of flash is its versatile workflow and production process. Flash uses vector graphics, which means that the graphics can be scaled to any size without losing clarity/quality.

Flash movies are graphics and animation for web sites. They consist primarily of vector graphics, but they can also contain imported bitmap graphics and sounds. Flash movies incorporate interactivity to permit input from viewers, and allow the users to create nonlinear movies that can interact with other Web applications [10]. Web designers use flash to create navigation controls, animated logos, long-form animations with synchronized sound, and even complete, sensory-rich web sites. Flash movies are compact, vector graphics, so they download rapidly and scale to the viewer's screen size. Flash allows the users to animate objects to make them appear to move across the stage and/or change their shape, size, color, opacity, rotation, and other properties. Flash animations are of two types, frame-by-frame animation, which creates a separate image for each frame and tweened animation, which creates the first and last frames of an animation and direct flash to create the frames in between. In Flash, we create a movie by drawing or importing artwork, arranging it on the Stage, and animating it with the Timeline.


00:04:35
PLAYING...
Show Index
Macromedia Flash MX Professional 2004 - [boy_model_static.fla*]
Amy Brool
chris geor
Edward Su

Forward in NYC?? Edward Sullivan: No pizza for you!! Jose Rodriguez : FF - NYC Todd Sanders: Argh chris georgenes: 4

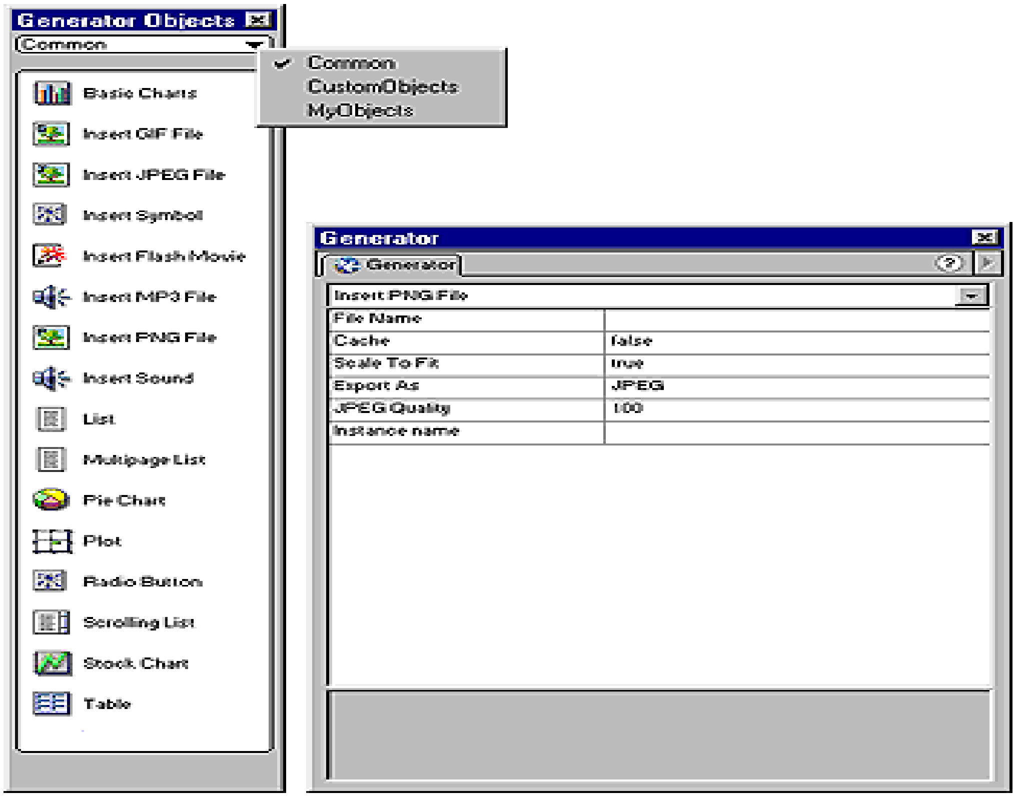
Figure 2.3 Developing Animations using Flash

Flash also allow us to create interactive movies, in flash we can use the keyboard or the mouse to jump to different parts of a movie, move objects, enter information in forms, and perform many other operations. When the movie is complete, we should export it as a Flash Player movie to be viewed in the Flash Player, or as a Flash stand-alone projector to be viewed with a self-contained Flash Player included within the movie itself. We can also generate static images like (JPEG, TIF etc) and animated gif's (GIF) using flash.

2.3 Jgenerator

Jgenerator is a web-server production application that can dynamically combine text, graphics, and sound to build rich media content and deliver the final product in a variety of animated or static formats [11]. Jgenerator also works similar to macromedia generator. It uses the same data format and same set of objects. Jgenerator combines the presentation templates that are generated using macromedia flash and generator objects (Figure 2.4). These templates contain placeholders for different multimedia elements like graphics, text, and sound. Jgenerator takes presentation templates and multimedia elements as the input and combines these two to generate the animation. The generator objects that are defined in the template will be replaced with the source content, the actual text files, databases, and graphics. The Jgenerator ultimately generates a Flash player movie, which can be played back in a browser.

This process of generating dynamic animated presentations using templates allows the users to use the same template to generate similar presentation using different data sources. Jgenerator frees designers from the tedium of updating flash movies each time they make a small change to the content. Jgenerator works in two modes, off-line mode (architecture shown in figure 2.5) and on-line mode (architecture shown in figure 2.6). In on-line mode, the Jgenerator runs under a web server and generates Flash movies by request from a browser.



Insert PNG File

File Name	
Cache	false
Scale To Fit	true
Export As	JPEG
JPEG Quality	100
Instance name	

Figure 2.4 Generator Objects

Online mode is used to generate dynamic animations. The animations are constructed on the fly based on a URL request from the user. Offline, the Jgenerator runs in command line and generates static Flash movies, which can be played back using flash player or used as required. Using offline mode, a designer can generate thousands of flash movies using just one template. Using online mode we can generate customized presentations for different users. Online processing is best suited for web sites where graphics need to be customized for each visitor to the Web site. Offline processing is best suited for Web sites where the data source changes infrequently. The following flowcharts demonstrate the process of using Jgenerator in two of the possible modes of operation: online, and offline.

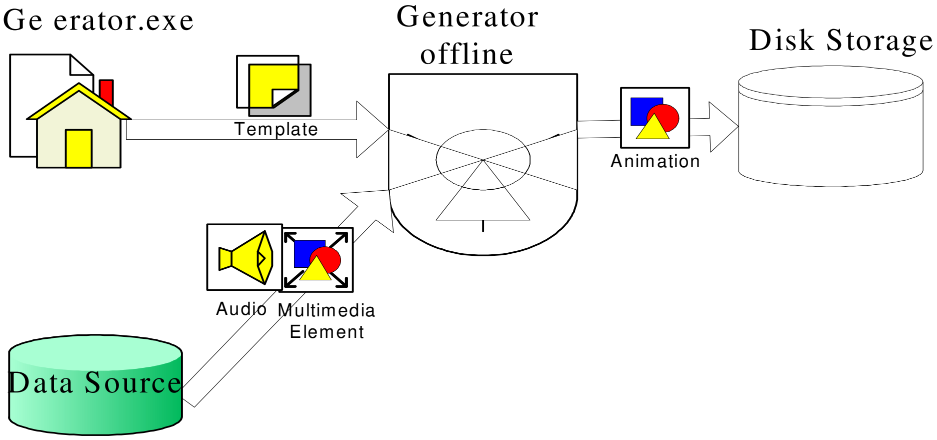


Figure 2.5 Generator Off-line Workflow

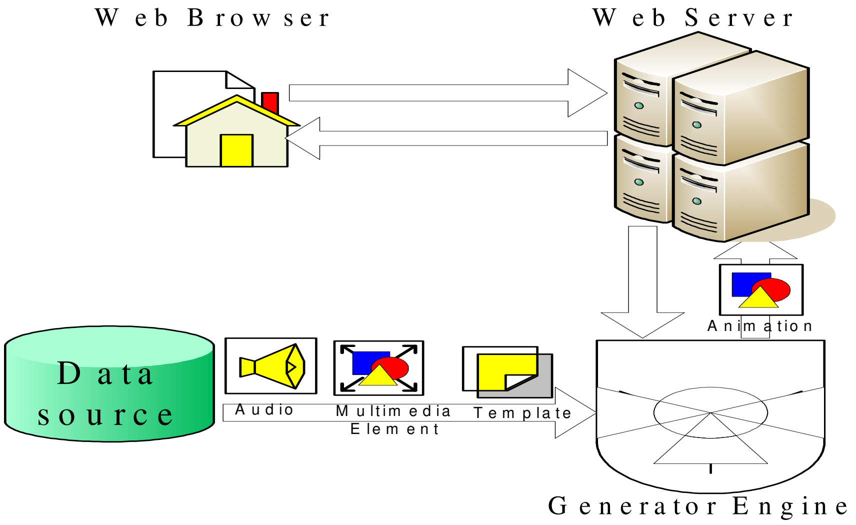


Figure 2.6 Jgenerator On-line Workflow

A Generator template has placeholders for text, graphics, and sound. Jgenerator can dynamically fill the placeholders with content from a data source to create a flash player movie. The data can be sport scores, stock quotes, news headlines, images, values for charts, and so on. Jgenerator supports different kinds of data sources. The data source can be a text file, text output returned by an HTTP or FTP request, a result set from a Structured Query Language (SQL) query, a user-created Java class or JavaScript. We use the Generator Environment Variable command to assign a data source to the timeline of the main movie or a movie clip. We can enter the file name for the data source in the dialog box (Figure 2.7) that opens when we click the Generator Environment Variable icon to the upper right of the timeline. In order to use more than one data source, we have to separate the data source names by semicolon. We can specify the data source format in two ways namely Name/Value and Column Name/Value. The main difference between the two formats is the number of values a variable can have. Name/Value variables can have only one value per variable, where as Column Name/Value variables can have more than one value. The Column Name/Value format is suited for multiple values and is required when we want to provide multiple sets (or records) of data to a command that takes multiple sets (Plot, Replicate, Scrolling List, Ticker, List, Chart, and so on).

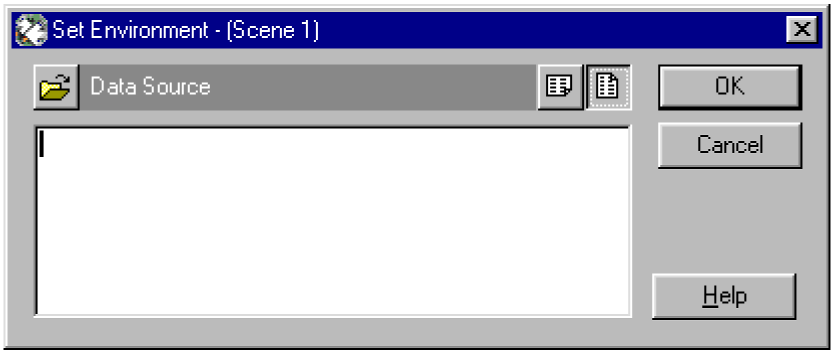


Figure 2.7 Defining Data Source for Generator Objects

Generator Data Source Format:

Name/Value
Name, Value
Var1, "Value1"
Var2, "Value2"
Var3, "Value3"

Column Name/Value
Var1, Var2, Var3

"Value1", "Value2", "Value3"
"Value1a", "Value2a", "Value3a"

Jgenerator can dynamically insert and modify movie clips and graphic symbols when processing a template. It can also dynamically modify the instances of the objects. We can replace a symbol with a different symbol from the library, or change the color of a symbol based on values provided when the template is processed. For example, if we are creating a temperature display, you can use the Set Tint command to change the temperature text color to red for warmer values or blue for cooler values. We can apply only one symbol command at a time to a symbol instance. If we need to use more than one symbol command, such as Transform and Set Alpha, place the original symbol instance inside a movie clip or graphic symbol. Nesting this way allows us to apply a symbol command to the original instance inside the new symbol and another command to the instance of the new symbol on the Stage.

2.4 Java Database Connectivity

JDBC (Java Database Connectivity) developed by Sun [12], is a Java API for connecting to and executing SQL statements in different Database Management Systems. JDBC includes a set of classes and interfaces written in Java. They are low level APIs, and are used to invoke SQL commands directly.

There are two kinds of models for database access: two-tier and three-tire. In the two-tier model, the client makes request to a DBMS associated with the server, through JDBC by sending SQL statements to the server and when the results are ready, the server will send the results back to the client. In the Three-tier model, instead of sending the request directly to the database server, the client sends requests to an intermediate server (termed application server), which applies some business logic, and then the SQL statements are sent to the database server by the application server. Finally, the result will be returned to the client through the application server. The advantage of the Three-tier model is that the business logic can be implemented in the application server, so when the business logic changes, only the program in the application server needs to be modified accordingly, and it is not necessary to update the program in the clients. Figures 2.8 and 2.9 show the architecture of two-tier and three-tier models. Because we are not incorporating any business logic and hence addition of a tier adds complexity without any gains for what we are trying to do, we choose the two-tier architecture for our data-mining project.

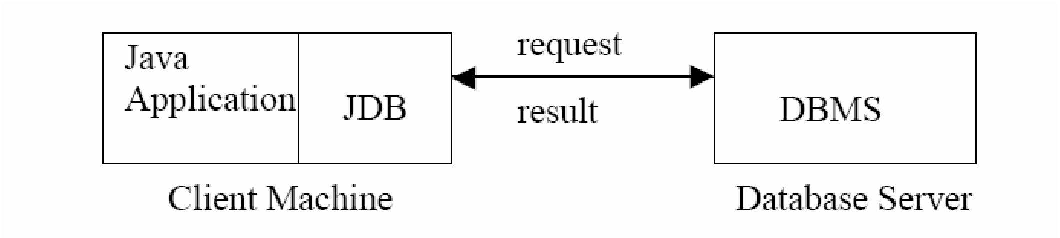


Figure 2.8 Architecture of Two-Tier Model

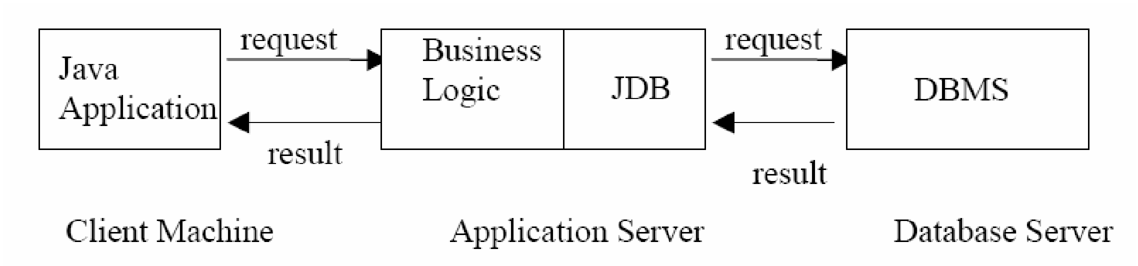


Figure 2.9 Architecture of Three-Tier Model

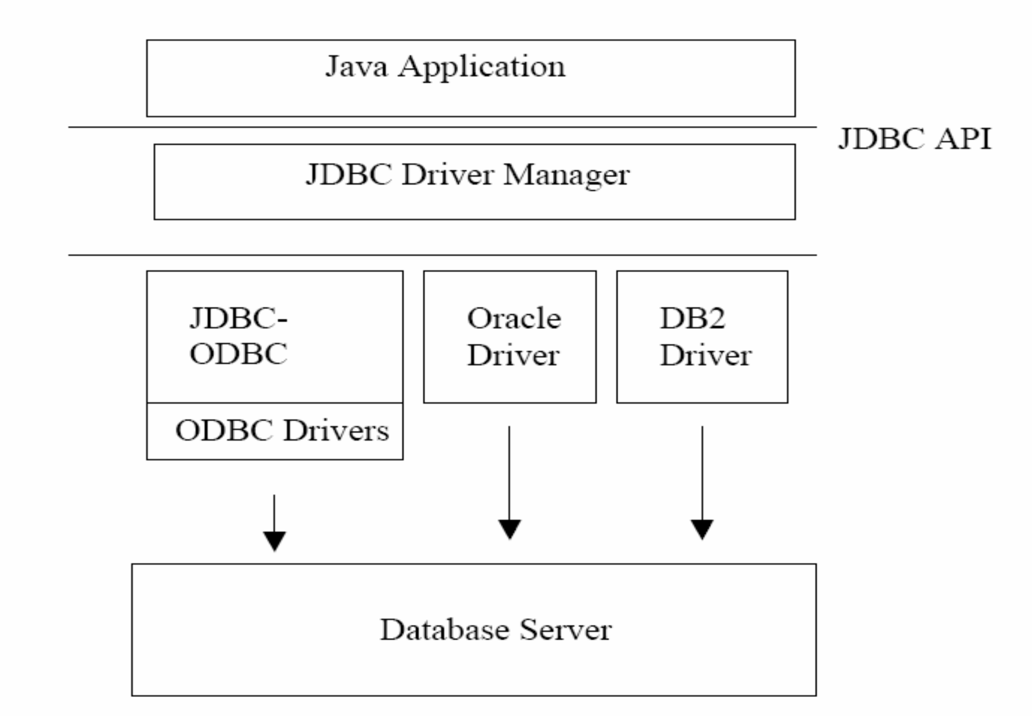


Figure 2.10 Architecture of JDBC

Figure 2.10 illustrates the architecture of JDBC. It has a driver manager that is responsible for choosing JDBC driver to be used for making a connection. JDBC-ODBC bridge driver is used to use ODBC through JDBC to access some of the less popular DBMS (such as MS Access) if JDBC drivers have not been implemented for them. The most important advantage of using JDBC compared to using ODBC is that JDBC drivers are written completely in Java, they are platform independent; ODBC is not completely platform independent and is not appropriate for direct use from Java.

2.5 MySQL

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, we need a database management system. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities or as parts of other applications.

A relational database system stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. MySQL is a relational database management system[13]. The SQL part of ``MySQL'' stands for ``Structured Query Language.'' SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. SQL-92' refers to the standard released in 1992, ``SQL:1999'' refers to the standard released in 1999, and ``SQL:2003'' refers to the current version of the standard. MySQL currently follows ``SQL:2003'' standard.

MySQL supports a number of data types in several categories: numeric types, date and time types, and string (character) types. BLOB stands for Binary Large Object. Both TEXT and BLOB are variable length types that store large amounts of data. They are similar to a larger version of VARCHAR. These types can store a large piece of data information, but they are also processed much slower. A BLOB is a binary large object that can hold a variable amount of data. The four BLOB types TINYBLOB, BLOB, MEDIUMBLOB and LONGBLOB differ only in the maximum length of the values they can hold. The four TEXT types TINYTEXT, TEXT, MEDIUMTEXT and LONGTEXT correspond to the four BLOB types and have the same maximum lengths and storage requirements. The only difference between BLOB and TEXT types is that sorting and comparison is performed in case-sensitive fashion for BLOB values and case-insensitive fashion for TEXT values. In other words, a TEXT is a case-insensitive BLOB. If you assign a value to a BLOB or TEXT column that exceeds the column type's maximum length, the value is truncated to fit. In most respects, you can regard a TEXT column as a VARCHAR column that can be as big as you like. Similarly, you can regard a BLOB column as a VARCHAR BINARY column. The differences between VARCHAR and BLOB are: You can have indexes on BLOB and TEXT columns with MySQL versions 3.23.2 and newer. Older versions of MySQL did not support this. There is no trailing-space removal for BLOB and TEXT columns when values are stored, as there is for VARCHAR columns. BLOB and TEXT columns cannot have DEFAULT values.

Data Types in MySQL:

CHAR	A fixed section from 0 to 255 characters long.
VARCHAR	A variable section from 0 to 255 characters long.
TINYTEXT	A string with a maximum length of 255 characters.
TEXT	A string with a maximum length of 65535 characters.
BLOB	A string with a maximum length of 65535 characters.
MEDIUMTEXT	A string with a maximum length of 16777215 characters.
MEDIUMBLOB	A string with a maximum length of 16777215 characters.
LONGTEXT	A string with a maximum length of 4294967295 characters.
LONGBLOB	A string with a maximum length of 4294967295 characters.

CHAPTER 3

DESIGN AND ARCHITECTURE

This chapter explains the design and architecture of the system. Section 3.1 briefly describes the Concept Map. Section 3.2 explains about Topic Map. Section 3.3 explains about Visualization Patterns and Presentation Templates. Section 3.4 explains the design and implementation of the database. Section 3.5 gives the overview of the system architecture.

3.1 Concept Map

Concept map is a technique for representing knowledge in graphs. A concept map provides the viewer with a graphical representation of a concept or idea. It is presented in a diagram format. Concept maps harness the power of our vision to understand complex information at-a-glance [14]. The primary function of the brain is to interpret incoming information to make meaning. It is easier for the brain to make meaning when information is presented in visual formats. This is why a picture is worth a thousand words. Concept mapping is the strategy employed to develop a concept map. A concept map consists of nodes or cells that contain a concept, item or question and links. The links are labeled and denote direction with an arrow symbol. The labeled links explain the relationship between the nodes. The arrow describes the direction of the relationship and reads like a sentence. Another feature of concept mapping is that it promotes higher order thinking, which allows meaningful learning to occur, and is more effective when done in a collaborative environment.

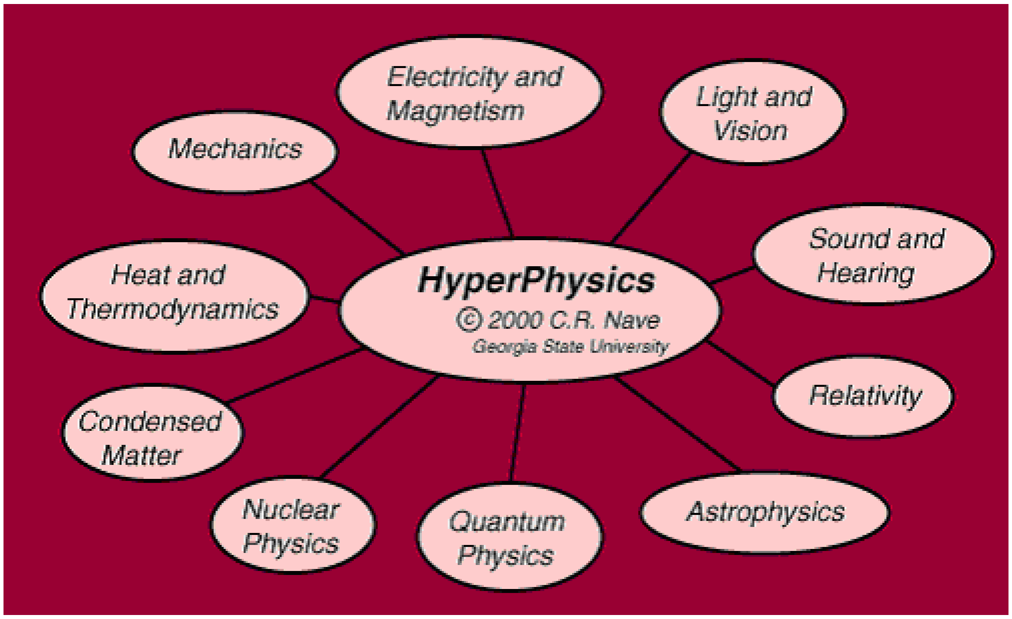


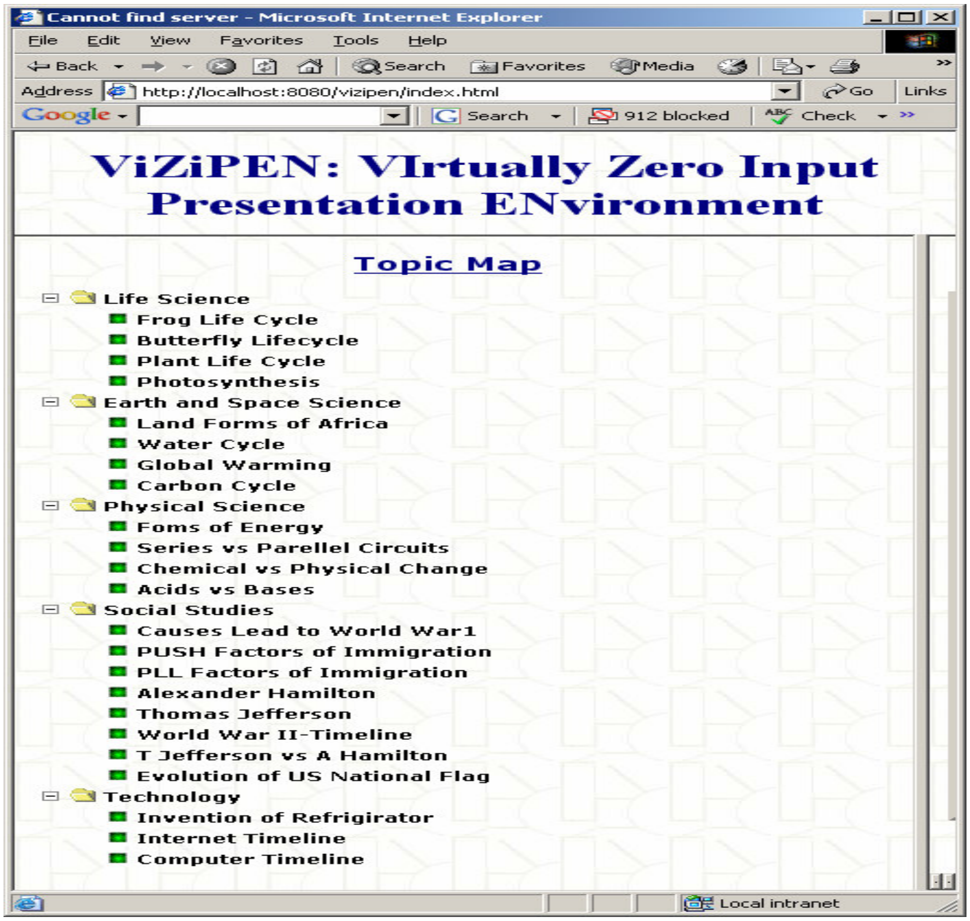
Figure 3.1 Hyper Physics Concept Map

Concept mapping is a technique that allows you to understand the relationships between ideas by creating a visual map of the connections. Concept maps allow us to see the connections between ideas we already have, to connect new ideas to knowledge that we have and to organize ideas in a logical but not rigid structure that allows future information or viewpoints to be included. Concept mapping looks like clustering (a type of free writing that is almost completely unstructured and that works by free association), but it goes one step further by revealing a clear relationship between the ideas that you're writing about. While concept mapping is more structured than prewriting, it is less structured and more flexible than formal outlining (which puts ideas in a sequence and organizes them by hierarchy or levels of importance), and so it allows you to see more complex relationships between ideas than just sequence and hierarchy.

3.2 Topic Map

Topic Maps were originally developed by a group in the SGML community to formalize the building of indices and lexicons. Topics are the basic building blocks in information visualization. Topic maps relate concepts by associations. A topic may also have a set of locations from which it may be accessed in some particular form. These locations are called topic occurrences. A topic may have a name, or it may have no name or more than one. Topics Maps also show lexicographical roots by building into the core model two forms of variants on the base name: display name and sort key. A topic, in its most generic sense, can be any thing, a person, an entity, a concept etc. Topic maps are a new ISO standard for describing knowledge structures and associating them with information resources.

In this thesis the topic map is generated based on the school curriculum. School curriculum contains various concepts that are related to each other. Since topic map is one of the most powerful representations of associations between related concepts, we used that approach. In this thesis the concepts are divided into different categories and each category contains a group of concepts in it. Concept maps have various levels of hierarchy. We can expand the number of levels as well as the number of groups based on how the concepts are related to each other. For this thesis we are considering the grade 8 science and social studies curricula as the knowledgebase, we classified the concepts in the knowledgebase into 5 categories. Each category contains a group of concepts that are closely associated as shown in figure.


Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/index.html Go Links
Google Search 912 blocked Check
Local intranet

ViZiPEN: VIrtually Zero Input Presentation ENvironment

Topic Map

- Life Science
  - Frog Life Cycle
  - Butterfly Lifecycle
  - Plant Life Cycle
  - Photosynthesis
- Earth and Space Science
  - Land Forms of Africa
  - Water Cycle
  - Global Warming
  - Carbon Cycle
- Physical Science
  - Foms of Energy
  - Series vs Parellel Circuits
  - Chemical vs Physical Change
  - Acids vs Bases
- Social Studies
  - Causes Lead to World War1
  - PUSH Factors of Immigration
  - PLL Factors of Immigration
  - Alexander Hamilton
  - Thomas Jefferson
  - World War II-Timeline
  - T Jefferson vs A Hamilton
  - Evolution of US National Flag
- Technology
  - Invention of Refrigirator
  - Internet Timeline
  - Computer Timeline

Figure 3.2 Curriculum Topic Map

3.3 Presentation Templates

In this presentation system we use pre-designed presentation templates that are generated using generator objects and flash. These templates have the placeholders where we can place the multimedia elements to generate the animated presentation. The template database stores these pre-designed animation templates. Section 3.3.1 discusses about visualization patterns, sections 3.3.2 explains about how to create presentation templates, where as section 3.3.3 explains about how to store the presentation templates in the template database.

3.3.1 Visualization Patterns

First and foremost, dynamic, interactive data visualization is about displaying abstract data, typically on a computer monitor. The data patterns focus on how to organize and visually represent the data in common visualization. The Decorated Data pattern addresses how to organize the common raw data, metadata, and various run-time states as the starting point of dynamic data visualization. The Visual Encoding pattern describes how the abstract data are mapped to visual forms for drawing.

Pattern is a method for capturing the knowledge underlying successful solutions to recurring architectural design problems. Dynamic visualization techniques [10] are widely used in various analytical visualization tasks including exploratory data analysis, process monitoring, and reporting results. While some of these tasks are very simple, many can also be quite complex. In either case, the development of software visualization systems to address these tasks is time consuming and burdensome. With the growing emphasis on information visualization in the scientific arena, new techniques are needed to improve the efficiency and effectiveness of the visualization development process.

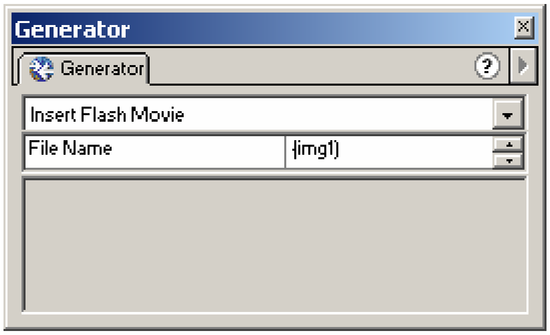
3.3.2 Designing Presentation Template

The presentation templates are built using macromedia flash and macromedia generator objects. Detailed explanation is given about macromedia flash in section 2.1. In general creating and deploying presentation template consists of these steps:

- Create Generator objects: Generator objects include charts, tables, lists, graphs and placeholders for flash player movies, symbol instances, and image files. We can also apply generator symbol commands to symbol instances. A generator template can contain more than one generator object. In this thesis we are using only JPEG, GIF and SWF generator objects provided by jgenerator.
- Selecting Generator Objects: Select generator objects from the generator object panel and place them on the stage in the flash movie. Add any animation effects to the generator object if required. (We can modify the instances like alpha, brightness, and tint for the generator objects

dynamically. We can also add any type of motion effects to the generator objects).

- Specify Name Variables For Generator Object. This variable is used to assign the multimedia element to the generator object. All variable names should be defined within {} parentheses (As shown in figure the file name should be defined as {img1}). In this thesis we are using TEXT, JPEG, GIF, MP3 and SWF templates that are provided by Jgenerator.



Insert Flash Movie

File Name	{img1}

Figure 3.3 Specifying Variables for Generator Object

- Specify Variables: Specify variables for the text objects if using any. Variables let us insert text into a generator template when content is created. We can also use variables as parameters for generator object properties, for example as parameters for most button and frame actions, or as frame labels. I am using variables for the description of object properties. In timeline I am also using the temporal variable that describes the time of the incident. I am also using variables for generator object properties such as name.

- Create data sources and assign them to main timelines or to specific objects. A data source can define text or graphics that are inserted in the template or it can contain values for generating objects such as a chart or a scrolling list. When we create a template, we assign to it data sources that contain values for variables in the template. For example, a data source can specify text, symbol names, or paths to external image files to be inserted when the template is processed. You can assign a data source to the main timeline of a template and to timelines of movie clips, as well as to specific Generator objects. A data source can be a comma-delimited text file, a URL that references a comma-delimited text file (either an actual file or a Web application that returns a text-only result), a result set from an SQL query passed through a JDBC/ODBC connection, a java class, or a native data source. (A native data source is data stored directly in the template rather than in an external file).
- View and Debug the Template. We can display a debugging report when you preview a generator template to check for errors and determine where we might need to make corrections to the template. See previewing, debugging, and exporting templates. Once the template is defines correctly, export the flash movie to a template and generate the (.swt) shockwave template file.

Once the template is generated successfully we need to add the template to the template database. A web-based user-interface (explained in section 3.3.3) is provided in this system to add the presentation template to the database.

3.4 Database Design and Implementation

The database is designed and developed using MySQL database. This database mainly used to store Animation Templates, Multimedia Elements and the conceptual information associated with the multimedia elements. Database consists of three tables namely imagedb, animdb and templatedb. Multimedia elements and animation templates are stored in the database in the form of byte array. Conversion of multimedia element or an animation template into a byte array is explained in detail in section 9.9. Imagedb table holds information about the image (.jpg and .gif) type multimedia elements. The animdb table holds information about the animation and video (.swf and .mov) type multimedia elements. The templatedb table holds information about the animation templates.

Imagedb table holds information about the image (.jpg and .gif) type multimedia elements and the conceptual information associated with them. Attributes are Name (Name of the image), Year (Applicable for images associated with Temporal Concepts), Description (description of the image), Fname (Name of the image file), Type (Multimedia element type), Grade (Grade to which the concept belongs), Subject

(Subject with which the concept is associated), Concept (Concept with which the multimedia element is associated), Sfname (Name of the audio file associated with the multimedia element), Melement (The byte array of multimedia elements) and Soundelement (Byte array of audio element). Combination of Name and Type is the primary key. Imagedb table attributes are shown in figure 3.2

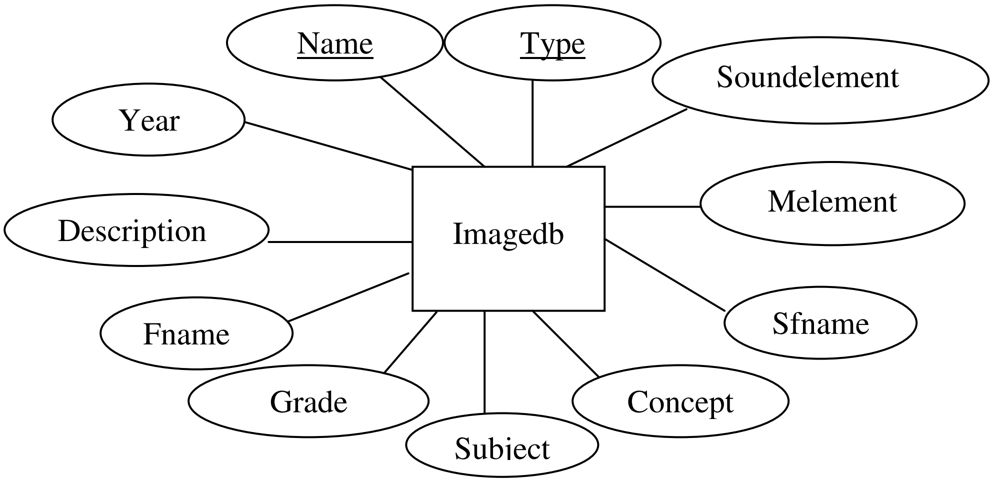


Figure 3.4 Imagedb Table

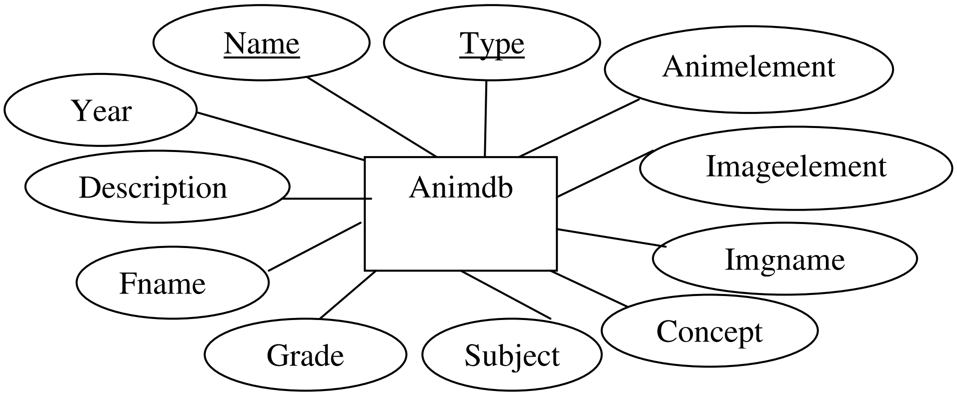


Figure 3.5 Animdb Table

Animdb table stores the information about animation and video type multimedia elements (.swf and .mov) and the conceptual information associated with these multimedia elements. The attributes are Name (Name of the image), Year (Applicable for images associated with Temporal Concepts), Description (description of the image), Fname (File name of the image file), Type (Multimedia element type), Grade (Grade to which the concept belongs), Subject (Subject with which the concept is associated), Concept (Concept with which the multimedia element is associated), Image (Name of the image file (a frame from the animation or video file), which represents the animation or movie), Imageelement (The byte array of image which represents the animation or video) and Animelement (Byte array of animation or video element). Combination of name and type is the primary key. The animdb table attributes are shown in figure 3.3.

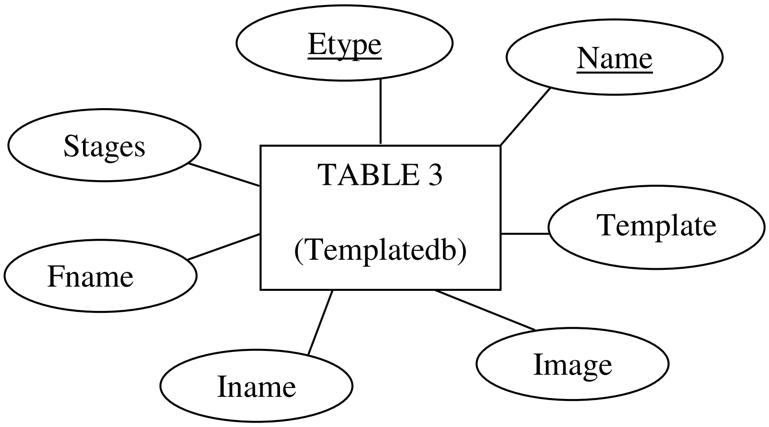


Figure 3.6 Templatedb Table

Templatedb table stores the information about animation templates (.swt files). The attributes are Name (Name of the template), Etype (Type of the elements that the template accepts), Stages (Number of staged in the template), Fname (File name of the template file), Iname (File name of the image file used to represent the template), Type (Multimedia element type), Grade (Grade to which the concept belongs), Subject (Subject with which the concept is associated), Concept (Concept with which the multimedia element is associated), Image (Name of the image file (a frame from the animation or video file), which represents the animation or movie), Imageelement (The byte array of image which represents the animation or video) and Animelement (Byte array of animation or video element). Combination of name and type is the primary key. The animdb table attributes are shown in figure 3.4.

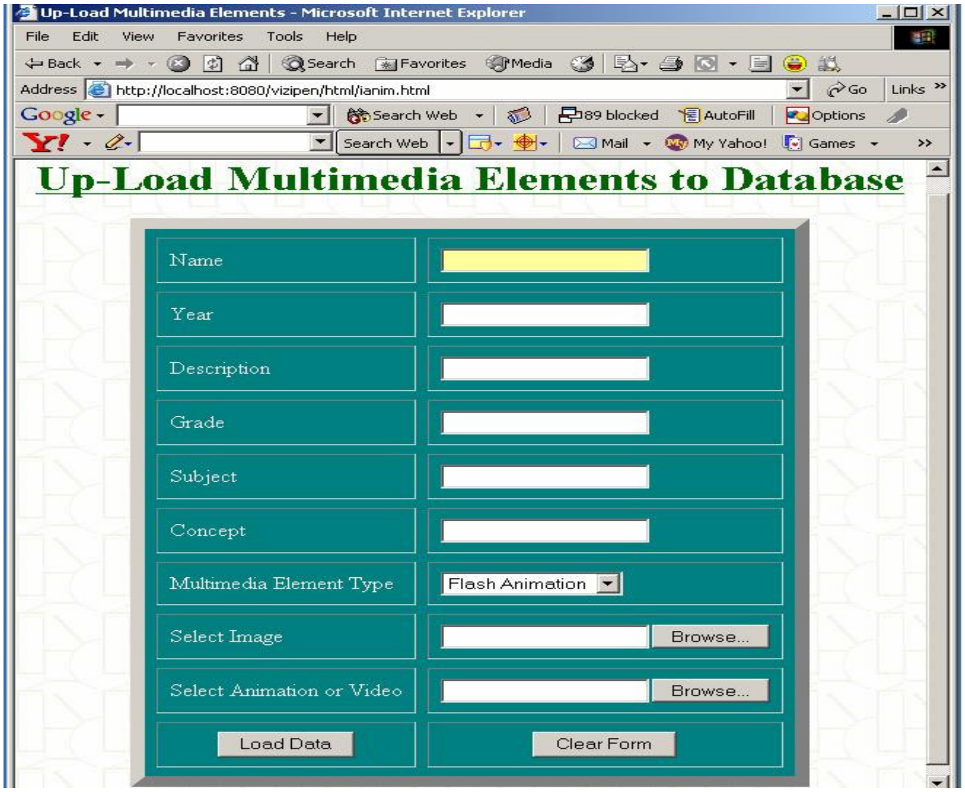
3.4.1 Uploading Multimedia Elements

A web-based user interface is provided to upload multimedia elements to the database. While uploading the elements the user has to provide the conceptual information associated with the multimedia elements and also the audio files associated with the elements. For the uploading process we classified the multimedia elements into two categories images (.jpg and .gif) and movie (.swf and .mov). Image type multimedia elements are JPG and GIF images. These images are directly viewable and so we do not need to store any frames for the representation of the multimedia elements.

Movie type multimedia elements are collection of frames having multimedia elements arranged in a functional order, so we are using a frame from the movie to represent it. We are also storing these frames in the database.

Figure 3.5 shows the GUI used to upload Image (JPG and GIF) type multimedia elements into the database. Name, Year, Description, Grade, Subject, Concept, Multiple Element Type, Select Image and Select Audio File fields are used to take inputs for the table columns Name, Year, Description, Grade, Subject, Type, Soundelement and Melement respectively. The values for columns Fname and Sfname will be extracted by using the complete paths of the multimedia element and Sound element respectively. From the complete path we are extracting the name of the multimedia element with its file type extension (Ex: finame.jpg). Once we get all input fields the string values will be added directly to the database, for the BLOB field we are converting the multimedia element to a byte array using the method described in section 3.4.2.

Figure 3.6 shows the GUI used to upload movie (SWF and MOV) type multimedia elements into the database. Name, Year, Description, Grade, Subject, Concept, Multiple Element Type, Select Image and Select Animation or Video are used to take inputs for the table columns Name, Year, Description, Grade, Subject, Type, Imgelement and Animelement respectively. The values for columns Imgname and Fname will be extracted by using the complete paths of the image and movie respectively. From the complete path we are extracting the name of the multimedia element with its file type extension (Ex: finame.swf). Once we get all input fields the string values will be added directly to the database, for the BLOB field we are converting the multimedia element to a byte array using the process is described in section 3.4.2.


Up-Load Multimedia Elements - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/html/ianim.html
Go Links
Google Search Web 89 blocked AutoFill Options
Y! Search Web Mail My Yahoo! Games
Up-Load Multimedia Elements to Database
Name
Year
Description
Grade
Subject
Concept
Select Image
Browse...
Select Animation or Video
Browse...
Load Data
Clear Form

Multimedia Element Type: Flash Animation

Figure 3.7 GUI to Upload (Image) Multimedia Elements to Database

3.4.2 Conversion of Multimedia Element to Byte Array

Multimedia elements (JPG, GIF, SWF, MOV and MP3 files) in the database are stored in the form of byte array, so are the animation templates. The process of converting images (multimedia element or animation templates) into a byte array is explained in the following section. This section describes the pseudo code (steps to be followed) to convert an image file into a byte array.

Conversion of Image to Byte Array:

```
Read the image file into a file input stream.
Create a database statement for mysql.
Construct “inset” query for adding new row to imagedb table.
Create prepared statement for “insert” query.
Set the binary stream in the prepared statement.
Execute the prepared statement.
Close the prepared statement.
```

The above method is used to convert any image file (JPG, GIF, SWF, MOV, MP3 and SWT) file into a byte array. First we read the file through a FileInputStream by using the complete path of the file as the input parameter. In the query we specify the value for the image element by using “?”, which means the value for this field is assigned to a variable which is defined in the next section. If we have more than one image element to be inserted we should use more than one variable (“?”). We are converting the FileInputStream to a binary stream in the next step. The binary stream will be added to the database, which will be stored as a BLOB (Binary Large Object) in the database. While retrieving the image from the database we need to do the reverse of this process, where we construct the image from the byte array. This is explained in the pseudo code described the following section.

Conversion of Image to Byte Array to Image:

Create file output stream with the file name of the images file in the database.

Create a byte array to write the image byte into it.

Get the binary stream of the image file from result set.

Write the binary stream from the image file to the file input stream.

Create a new output stream with max file size.

Create a new byte array.

Read all the bytes from input stream and write them into output stream.

Convert the file output stream to a byte array.

Close file input stream.

Close file output stream.

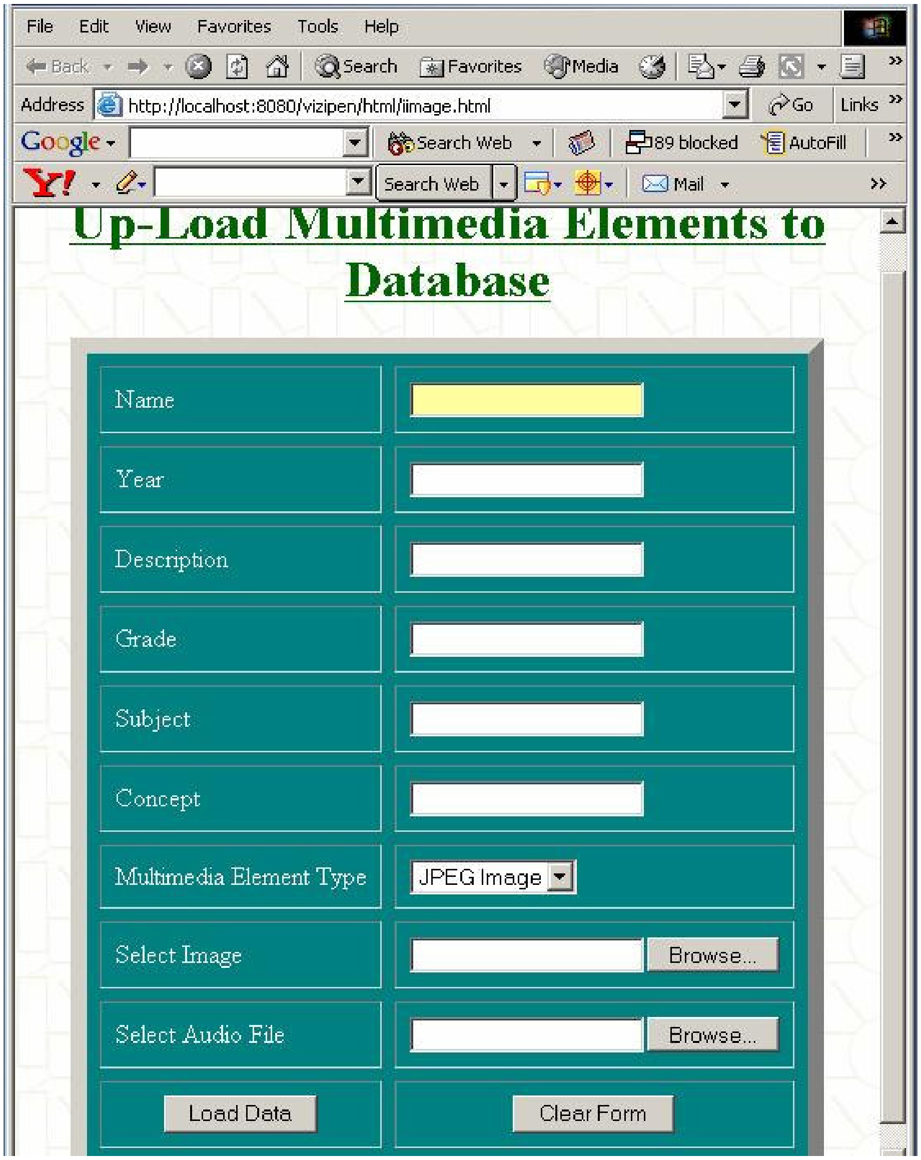
Flush file output stream.

Write the file output stream into the byte array.

Convert the byte array to image.

Close file output stream.

In order to reconstruct the multimedia element from the byte array the above method is used. In this process we get the byte array from the result set. We are reading the byte array into a byte array output stream and converting this to a byte array. Then we are writing the byte array into a file output stream. After this we are creating an empty image file with the file name that we retrieved from the database and writing the file output stream into that file and saving that file on the disk. These files are saved under a temporary directory onto the disk and will be removed at regular intervals as part of maintenance work.

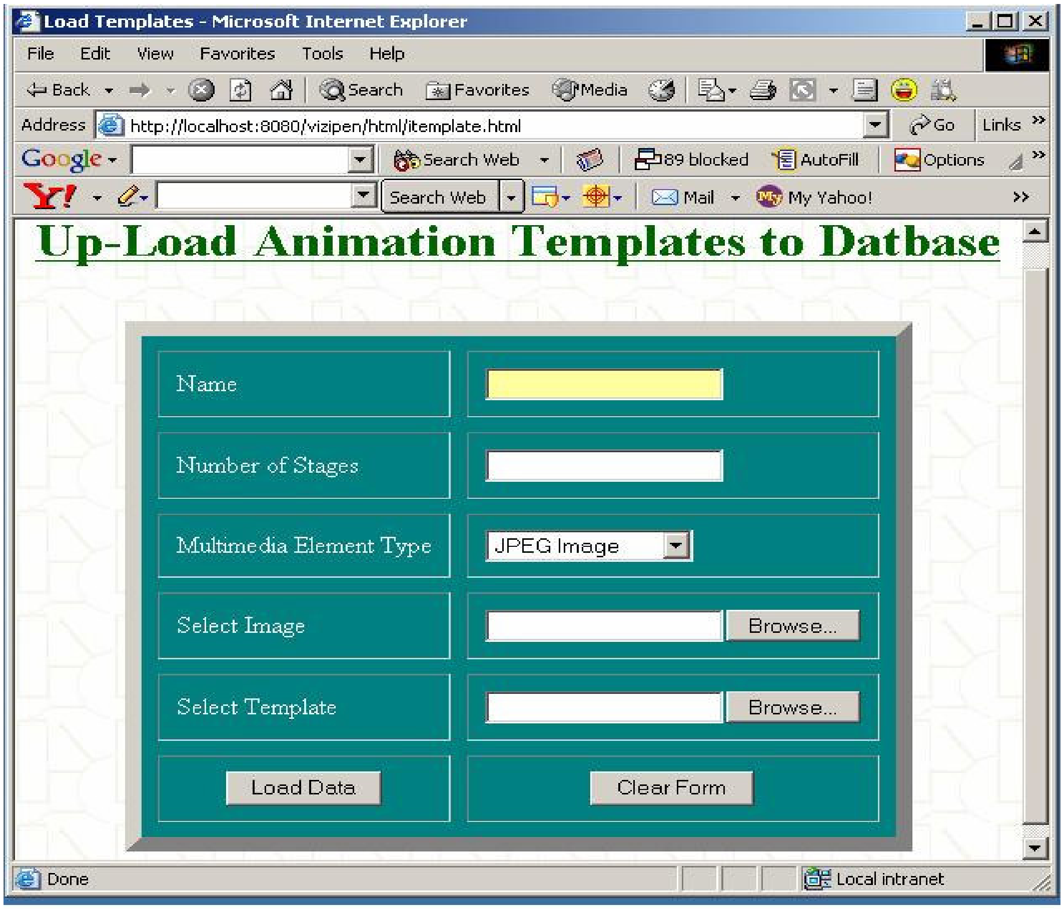


File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/html/iimage.html Go Links
Google Search Web 189 blocked AutoFill
Y! Search Web Mail

Up-Load Multimedia Elements to Database

Name	
Year	
Description	
Grade	
Subject	
Concept	
Multimedia Element Type	JPEG Image
Select Image	Browse...
Select Audio File	Browse...
Load Data	Clear Form

Figure 3.8 GUI to upload (Move) Multimedia Elements to Database


Load Templates - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/html/itemplate.html
Go Links
Search Web 89 blocked AutoFill Options
Search Web Mail My Yahoo!
Browse...
Browse...
Load Data
Clear Form
Done
Local intranet

Up-Load Animation Templates to Datbase

Name	
Number of Stages	
Multimedia Element Type	JPEG Image
Select Image	
Select Template	

Figure 3.9 GUI to Upload Animation Templates to Database

3.4.3 Uploading Animation Templates

Figure 3.14 shows the GUI used to upload Presentation templates (.swt) to the database. Name, Number of Stages, Element Type, Select Image and Select Template are used to take inputs for the table columns Name, Stages, Type, Subject, Type, Image and Template respectively. The value for column Fname and Iname will be extracted by using the complete path of the image and the template. From the complete path we are extracting the name of the multimedia element with its file type extension (Ex: finame.jpg). Once we get all input fields the string values will be added directly to the database, for the BLOB filed we are converting the multimedia element to a byte array using the process described in section 3.4.2.

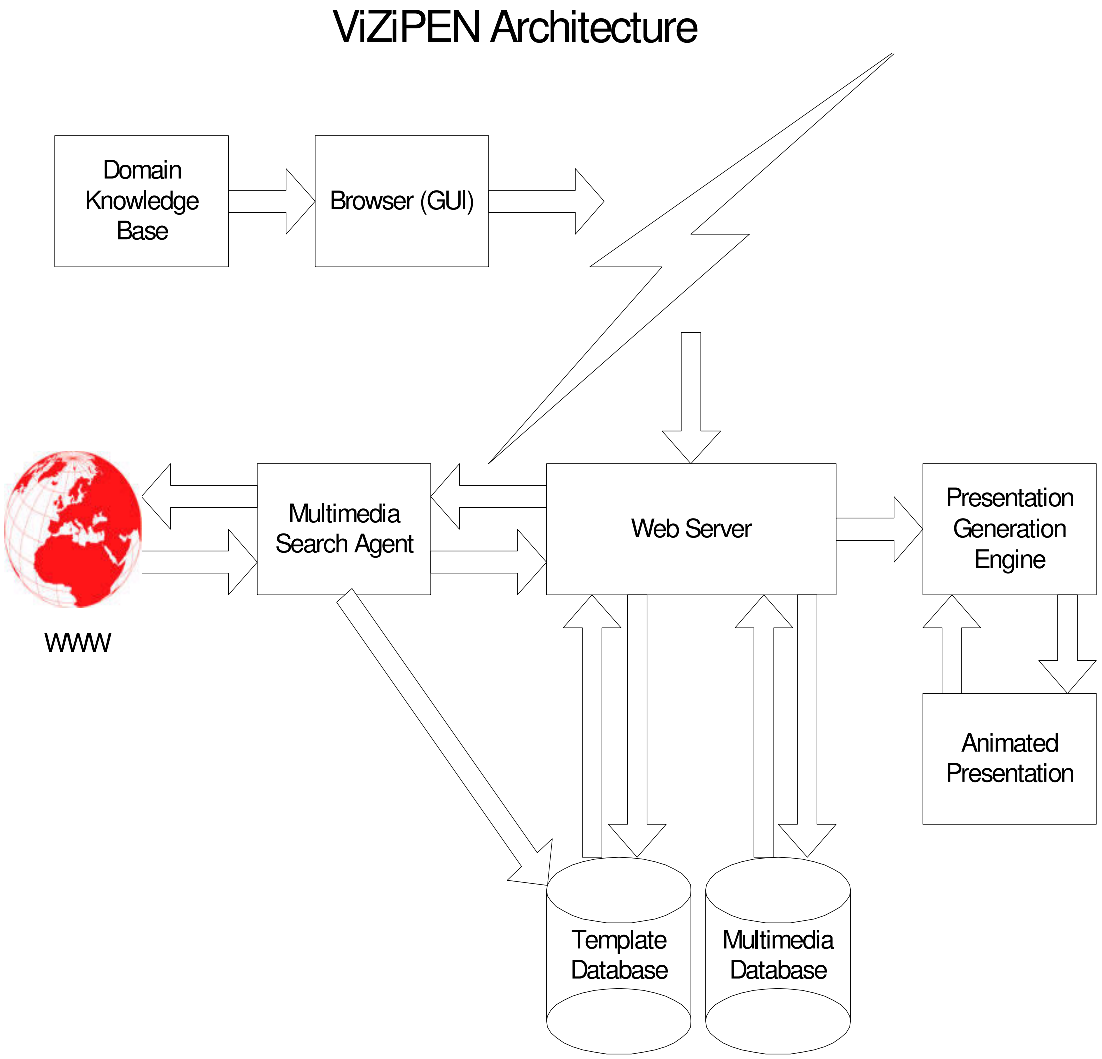


Figure 3.10 ViZiPEN System Architecture

3.5 System Architecture

The ViZiPEN system is designed around a web / application server that acts as a mediator for six other components as illustrated in figure 3.8.

These components are:

- Presentation Template Database: This database consists of pre-designed animation templates with replaceable component objects such as images, animations and text.
- Presentation Generation Engine: This engine (the "Generator" in the sequel) enables automated generation of presentations by replacing designated objects within the presentation or their properties. Generator is the server side software for automatically delivering dynamic, on-the-fly, personalized web graphics immediately or at scheduled times. The generator takes the animated template and inserts the content objects (text, image, animation, video etc) into the corresponding places in the template. The completed animation is then sent to the web-server to be delivered to the user.
- Multimedia Web Search Agent: This is an information retrieval and extraction agent for identifying multimedia content elements on the web and associating semantic information with them. The elements and their associated information are then inserted into an object-relational database.

- Multimedia Database: This database consists of multimedia elements such as images, sound bytes, video clips, graphic objects and animations that can be inserted into animated presentations. Each element has a number of semantic meta-data attributes that enable intelligent querying of these elements. Example attributes include content topic structure, date of creation etc.
- Domain Knowledge Base: The domain knowledge base enables concept expansion and automatic multimedia element collection. Concepts from domain knowledge sources such as digitized curricula are captured with knowledge representation methods. Starting with concepts, related entities of interest are identified via concept expansion. Each multimedia entity is associated with a number of concepts.
- Web-based User Interface: The user interface provides two main functions: It enables the user to identify presentation templates and customize them with multimedia elements from the database. It also facilitates the user for the addition of new presentation templates as well as multimedia elements.

CHAPTER 4

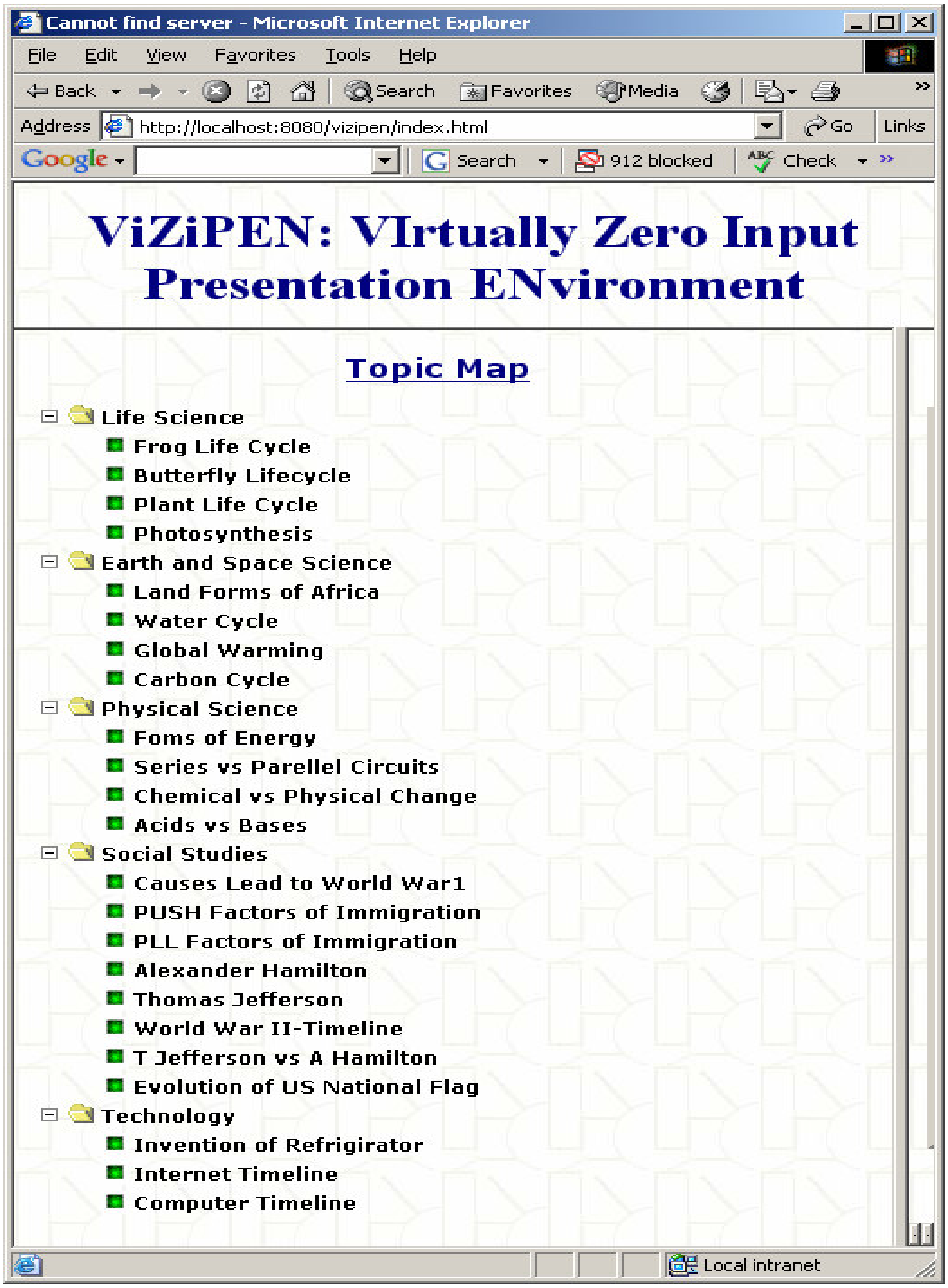
IMPLEMENTATION

This chapter gives the overview of vizipen system and how it has been implemented. Figure 4.1 shows the Schematic diagram of vizipen system components. It consists of six modules along with the database. Module 1 is the curriculum topic map. The topic map is used to represent the school curriculum. Module 2 is the presentation template database, which stores the pre-designed presentation templates. Module 3 is multimedia element database, which stores the multimedia elements in the database. Module 4 is the multimedia search agent, the coordinator component that interacts with the database and the curriculum. This module queries the database for extracting the multimedia elements associated with a particular concept in the curriculum. Module 5 is the GUI, which coordinates with all other modules and facilitates the user with the functionality. Module 6 is the presentation generation engine, which is responsible for the overall functionality of the system. This module takes input from the GUI, extracts multimedia elements and presentation templates from the database as per the user selection and hence generates the animated presentation by combining all these components.

4.1 Curriculum Topic Map

Curriculum topic map represents the digitized school curriculum. The school curriculum is represented in the form of a topic map. The topic map consists of curriculum for different grades and different subjects for each grade. Each subject is further classified into some topics or subtopics, each topic or subtopic contains a number of concepts associated with them. Some associations relate the concepts and topics. Figure 4.1 represents the topic map curriculum for grade 8. In this thesis we are using only grade 8 curriculums and so prepared the Topic map for grade 8 curriculums. The topic map consists of different subject as the abstract concepts at a higher level, in the next level each subject consists of a number of concepts. These concepts are associated with some multimedia elements, which are used to illustrate these concepts.

The above figure shows the topic map for Grade 8 curriculum. The topic map contains 5 subjects namely Life Science, Earth and Space Science, Physical Science, Social Studies and Technology. These subjects are associated with some concepts. Each concept is associated with some multimedia elements. The user can select any of these concepts by clicking on them and the multimedia search agent will get the corresponding multimedia elements. The user selects the multimedia elements from these and put them in the placeholders of the templates, thus by giving these elements as inputs to the presentation generation engine.



ViZiPEN: VIrtually Zero Input Presentation ENvironment

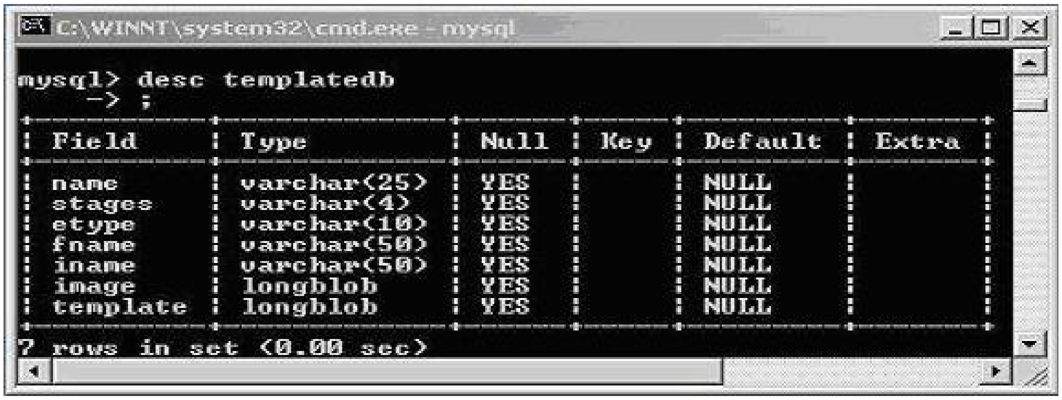
Topic Map

- ⊟ Life Science
  - 🟩 Frog Life Cycle
  - 🟩 Butterfly Lifecycle
  - 🟩 Plant Life Cycle
  - 🟩 Photosynthesis
- ⊟ Earth and Space Science
  - 🟩 Land Forms of Africa
  - 🟩 Water Cycle
  - 🟩 Global Warming
  - 🟩 Carbon Cycle
- ⊟ Physical Science
  - 🟩 Foms of Energy
  - 🟩 Series vs Parellel Circuits
  - 🟩 Chemical vs Physical Change
  - 🟩 Acids vs Bases
- ⊟ Social Studies
  - 🟩 Causes Lead to World War1
  - 🟩 PUSH Factors of Immigration
  - 🟩 PLL Factors of Immigration
  - 🟩 Alexander Hamilton
  - 🟩 Thomas Jefferson
  - 🟩 World War II-Timeline
  - 🟩 T Jefferson vs A Hamilton
  - 🟩 Evolution of US National Flag
- ⊟ Technology
  - 🟩 Invention of Refrigirator
  - 🟩 Internet Timeline
  - 🟩 Computer Timeline

Figure 4.1 Topic Map for Grade 8 Curriculum

4.2 Presentation Template Database

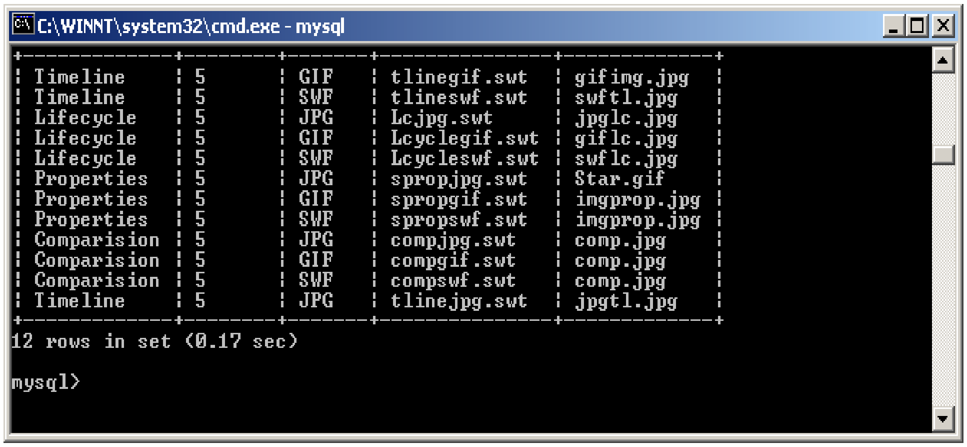
The presentation template database consists of animated presentation templates (.swt files) developed using Macromedia Flash and Macromedia Generator. These templates are stored in templatedb table under vizipen database. The presentation templates are stored in the database in the form of binary data (byte array) as explained in section 3.4.1. In this thesis we developed presentation templates only for 4 types of concept maps. The presentation templates that we have developed for this thesis are homogeneous (the template accepts only one type of multimedia elements – JPG or GIF or SWF or MOV). With this constraint we developed one template for each type of multimedia element for every concept. The database contains a total of 12 templates 3 for each concept map representation. Based on user selection for the type of multimedia element we query the database for the corresponding presentation and retrieve it from the database.



```
mysql> desc templatedb
    -> ;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| name     | varchar(25) | YES  |     | NULL    |       |
| stages   | varchar(4)  | YES  |     | NULL    |       |
| etype    | varchar(10) | YES  |     | NULL    |       |
| fname    | varchar(50) | YES  |     | NULL    |       |
| iname    | varchar(50) | YES  |     | NULL    |       |
| image    | longblob    | YES  |     | NULL    |       |
| template | longblob    | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Figure 4.2 Templatedb Table Design

Presentation templates are generated for timeline, life cycle, properties and comparison concept maps. These presentation templates are stored on the templatedb table as shown in Figure 4.3. The table has 12 rows and each row represents a template. In this table we are showing only the first 5 columns because the last two columns are BLOB data types.



```
+-------------+--------+-------+---------------+-------------+
| Timeline    | 5      | GIF   | tlinegif.swt  | gifimg.jpg  |
| Timeline    | 5      | SWF   | tlineswf.swt  | swftl.jpg   |
| Lifecycle   | 5      | JPG   | Lcjpg.swt     | jpglc.jpg   |
| Lifecycle   | 5      | GIF   | Lcyclegif.swt | giflc.jpg   |
| Lifecycle   | 5      | SWF   | Lcycleswf.swt | swflc.jpg   |
| Properties  | 5      | JPG   | spropjpg.swt  | Star.gif    |
| Properties  | 5      | GIF   | spropgif.swt  | imgprop.jpg |
| Properties  | 5      | SWF   | spropswf.swt  | imgprop.jpg |
| Comparision | 5      | JPG   | compjpg.swt   | comp.jpg    |
| Comparision | 5      | GIF   | compgif.swt   | comp.jpg    |
| Comparision | 5      | SWF   | compswf.swt   | comp.jpg    |
| Timeline    | 5      | JPG   | tlinejpg.swt  | jpgtl.jpg   |
+-------------+--------+-------+---------------+-------------+
12 rows in set (0.17 sec)

mysql>
```

Figure 4.3 Templatedb Table

Timeline: This concept map can be used to represent temporal concepts (Concepts that are associated with time). For example we can represent the incidents in World War II by using the time line concept map. As shown in figure 4.4 the concept map contains a horizontal bar, which represents the time associated with the multimedia element. The presentations shows the events by showing up the images related to the event at different times by populating the images, it also plays the sound in the background.

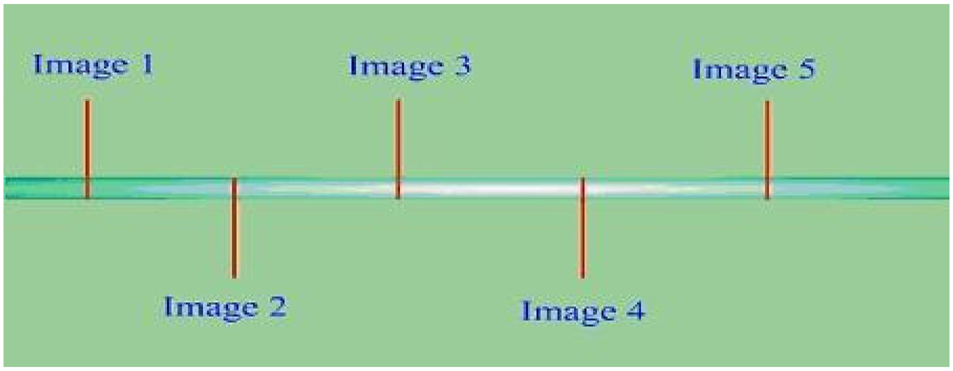


Figure 4.4 Timeline Concept Map

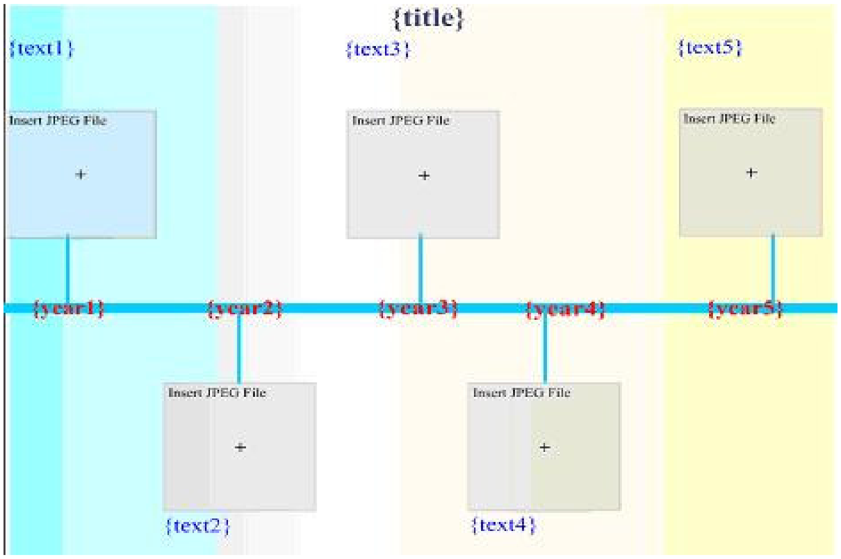


Figure 4.5 Timeline template

Figure 4.5 shows the timeline template for JPG image type multimedia elements. The template has some empty rectangles with text saying insert JPEG File; these are the generator objects for JPEG images. When the user places the multimedia element in the database these placeholders will be replaced by actual multimedia elements. The variables (text in between {}) are used to represent the description associated with each multimedia element. The description parameter associated with the multimedia element will be used to replace these variables. The sound objects are not shown in the figure since they are hidden; the mp3 files that are associated with each multimedia element replace these objects.

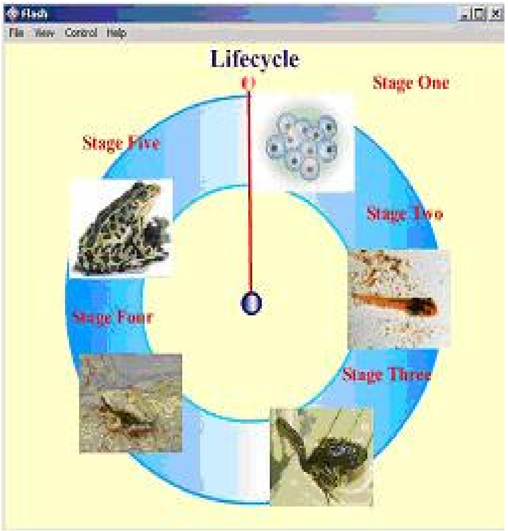


Figure 4.6 Lifecycle Concept Map

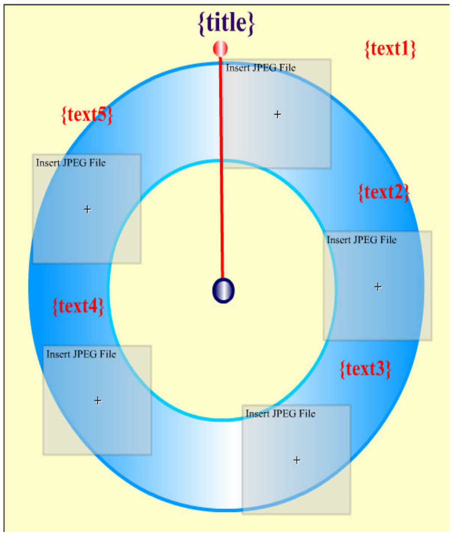


Figure 4.7 Lifecycle Template

Lifecycle: This concept map can be used to represent cyclic concepts (Concepts that follow a cycle). For example we can represent the life cycle of a frog by using the life cycle concept map. As shown in figure 4.6 the concept map contains a cyclic representation, which associates the events in a cyclic fashion. The presentations shows the events by showing the images related to the event at different stages in a cyclic order by populating the images, it also plays the sound in the background. The template contains placeholders for images and description for the images as shown in figure 4.7.

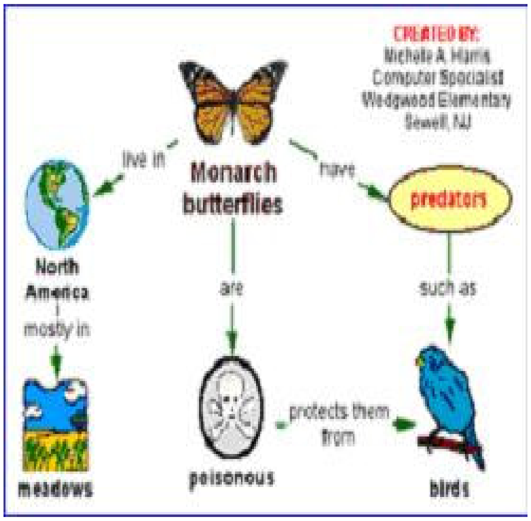


Figure 4.8 Properties Concept Map

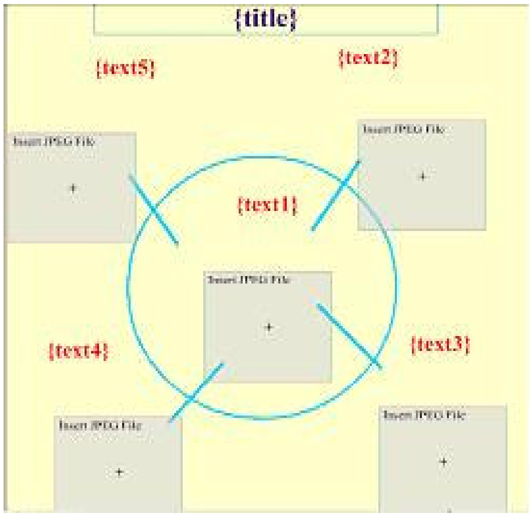


Figure 4.9 Properties Template

Properties: This concept map can be used to represent concepts describing properties of any object. For example we can represent the properties of metals in periodic table by using the properties concept map. As shown in figure 4.8 the concept map contains a property description representation, which associates the events in extended fashion. The presentations shows the events by showing up the images related to the event in different directions by populating the images, it also plays the sound in the background. Figure 4.9 shows how the template looks like when there is no multimedia elements filled in the placeholders.

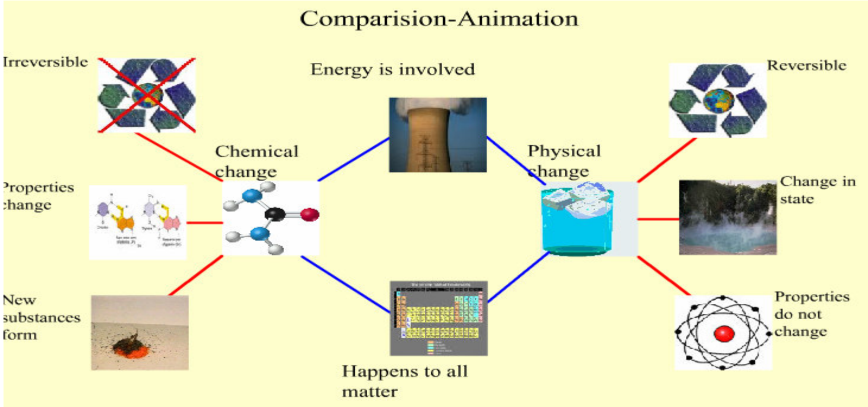


Figure 4.10 Comparison Concept Map

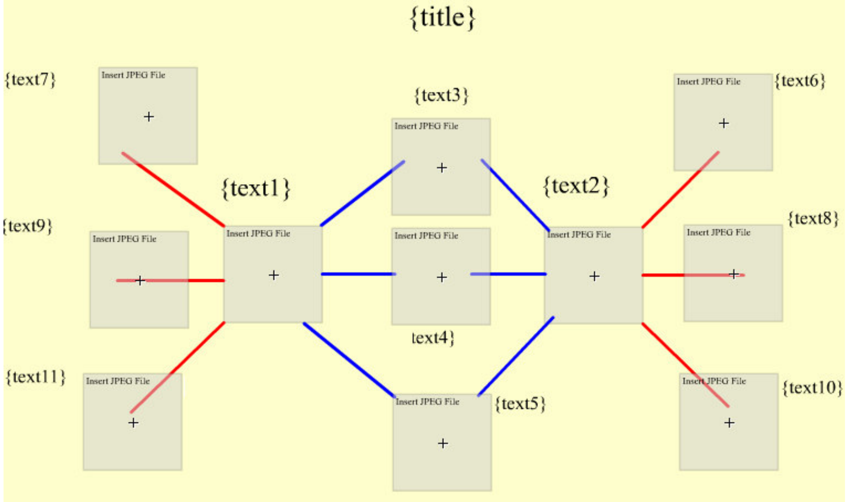
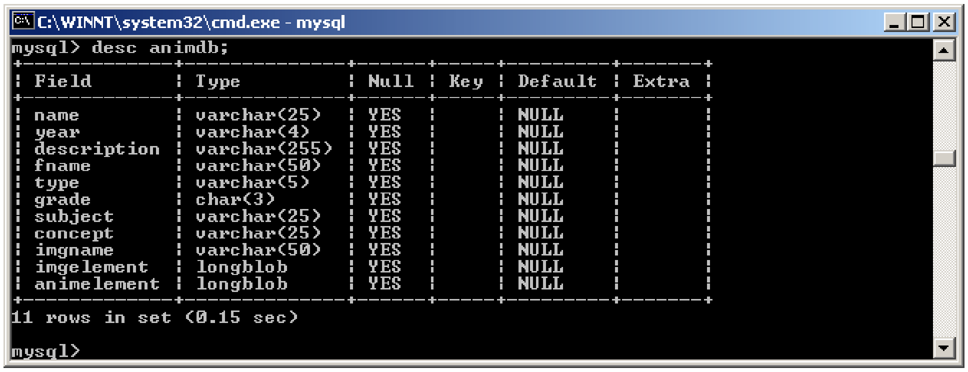


Figure 4.11 Comparison Template

Comparison: This concept map can be used to compare concepts with some common and some distinct properties. As show in the figure 4.10 the concept map will have two separate entities representing each concept, common properties are linked to both the entities where as individual properties of the concepts are linked to the corresponding entities. This is a very useful concept map because this gives a clear picture of how two concepts differ or how they are related to each other. Figure 4.11 shows the comparison template. For any concept map the template will be exactly similar to the concept map without the multimedia elements in the placeholders.

4.3 Multimedia Element Database

The multimedia elements that are used in the placeholders of the templates are stored in animdb and imagedb tables' under vizipen database. Animations and images are treated as two different types of multimedia elements based on their nature. Animation / movie contains more than one frame where as an image is equivalent to a frame of an animation or a movie. So when the user selects the concept we are presenting him a single frame from the animation or movie, which he can drag and drop into the placeholder on the template.

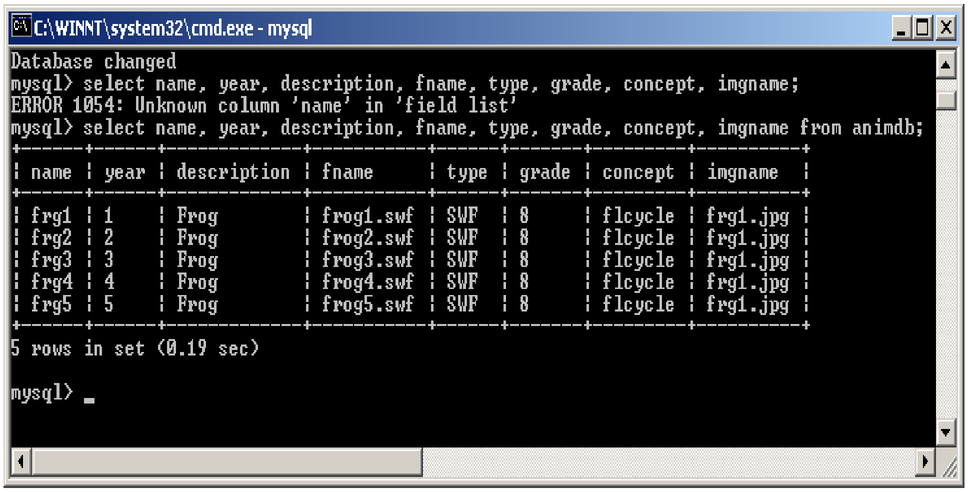


```
mysql> desc animdb;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| name        | varchar(25)  | YES  |     | NULL    |       |
| year        | varchar(4)   | YES  |     | NULL    |       |
| description | varchar(255) | YES  |     | NULL    |       |
| fname       | varchar(50)  | YES  |     | NULL    |       |
| type        | varchar(5)   | YES  |     | NULL    |       |
| grade       | char(3)      | YES  |     | NULL    |       |
| subject     | varchar(25)  | YES  |     | NULL    |       |
| concept     | varchar(25)  | YES  |     | NULL    |       |
| imgname     | varchar(50)  | YES  |     | NULL    |       |
| imgelement  | longblob     | YES  |     | NULL    |       |
| animelement | longblob     | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
11 rows in set (0.15 sec)

mysql>
```

Figure 4.12 Animdb Table Design

Animdb stores multimedia elements like flash animations (of type .swf) and quicktime movie (of type .mov) files. These two types' elements play a movie or an animation. As shown in figure 4.12 animdb table stores a frame of the animation or movie in another column for representation purpose. Imageelement column stores a single frame of the animation or movie in the binary format. When a user wants to drag or drop a flash / quicktime movie into a presentation template, he will drag and drop the image that represents the animation. Animdb table contains name, year, description, filename, element type, grade, subject, concept, name of the image, image representation (single frame of animation) and animation fields. The last two fields are BLOB type and store an image and an animation in binary format. Figure 4.13 shows the elements stored in the animdb table. In this table we are not showing the BLOB type elements.

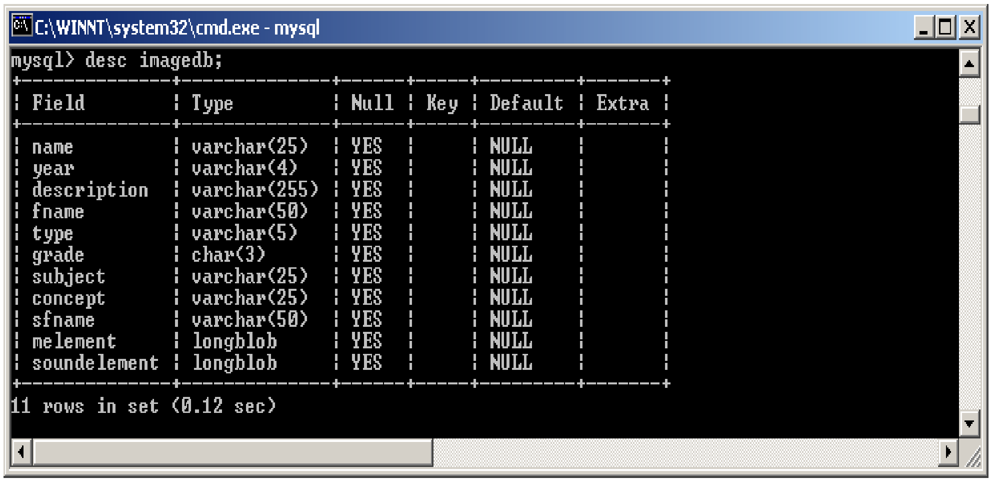


```
Database changed
mysql> select name, year, description, fname, type, grade, concept, imgname;
ERROR 1054: Unknown column 'name' in 'field list'
mysql> select name, year, description, fname, type, grade, concept, imgname from animdb;
+------+------+-------------+-----------+------+-------+---------+----------+
| name | year | description | fname     | type | grade | concept | imgname  |
+------+------+-------------+-----------+------+-------+---------+----------+
| frg1 | 1    | Frog        | frog1.swf | SWF  | 8     | flcycle | frg1.jpg |
| frg2 | 2    | Frog        | frog2.swf | SWF  | 8     | flcycle | frg1.jpg |
| frg3 | 3    | Frog        | frog3.swf | SWF  | 8     | flcycle | frg1.jpg |
| frg4 | 4    | Frog        | frog4.swf | SWF  | 8     | flcycle | frg1.jpg |
| frg5 | 5    | Frog        | frog5.swf | SWF  | 8     | flcycle | frg1.jpg |
+------+------+-------------+-----------+------+-------+---------+----------+
5 rows in set (0.19 sec)

mysql> _
```

Figure 4.13 Animdb Table

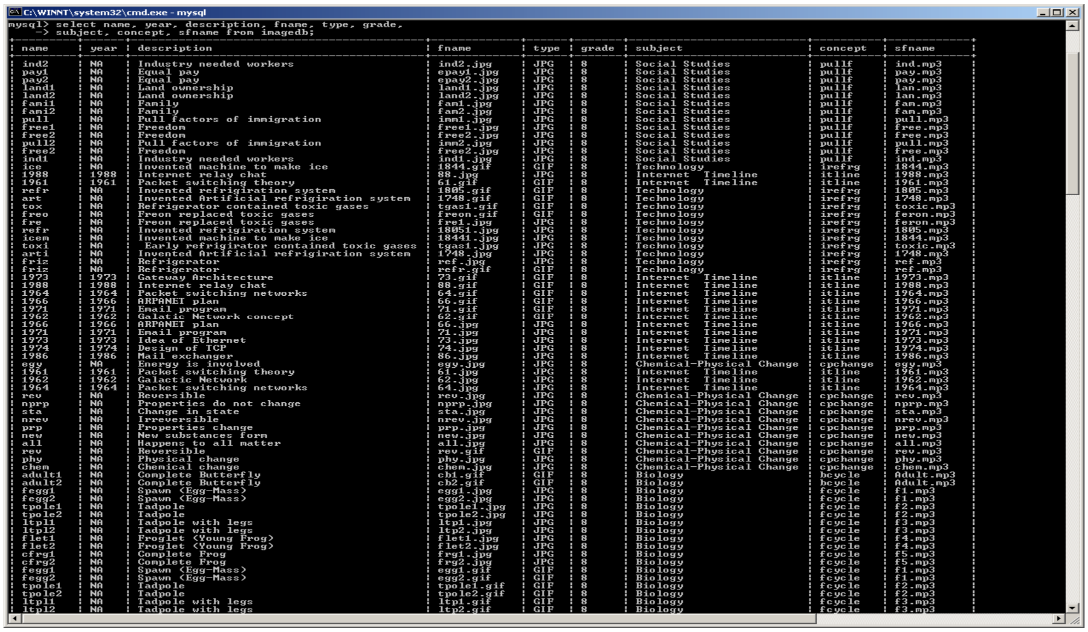
Image type (JPG / SWF files) multimedia elements are stored in imagedb table under vizipen database. As shown in figure 4.14 imagedb table stores image files and audio file in the binary format. The columns melement and soundelement are used to store the image and audio files in binary format. Other columns store the other properties like subject, grade, concept etc of the multimedia element. Type field is used to identify the type of the multimedia element. In this thesis we are supporting 2 types of images JPG and GIF so type filed contains the value JPG or GIF based on the type of the images that we are storing. Melement and soundelement are BLOB data types, which are used to store the binary format of image and sound files.



```
mysql> desc imagedb;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| name         | varchar(25)  | YES  |     | NULL    |       |
| year         | varchar(4)   | YES  |     | NULL    |       |
| description  | varchar(255) | YES  |     | NULL    |       |
| fname        | varchar(50)  | YES  |     | NULL    |       |
| type         | varchar(5)   | YES  |     | NULL    |       |
| grade        | char(3)      | YES  |     | NULL    |       |
| subject      | varchar(25)  | YES  |     | NULL    |       |
| concept      | varchar(25)  | YES  |     | NULL    |       |
| sfname       | varchar(50)  | YES  |     | NULL    |       |
| melement     | longblob     | YES  |     | NULL    |       |
| soundelement | longblob     | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
11 rows in set (0.12 sec)
```

Figure 4.14 Imagedb Table Design

As shown in figure 4.15, imagedb table contains multimedia elements related to various concepts. The image does not show the BLOB type multimedia elements.



```
mysql> select name, year, description, fname, type, grade,
    -> subject, concept, sfname from imagedb;
+------------+------+-------------------------------------------+---------------+------+-------+--------------------------+----------+-----------+
| name       | year | description                               | fname         | type | grade | subject                  | concept  | sfname    |
+------------+------+-------------------------------------------+---------------+------+-------+--------------------------+----------+-----------+
| ind2       | NA   | Industry needed workers                   | ind2.jpg      | JPG  | 8     | Social Studies           | pullf    | ind.mp3   |
| pay1       | NA   | Equal pay                                 | epay1.jpg     | JPG  | 8     | Social Studies           | pullf    | pay.mp3   |
| pay2       | NA   | Equal pay                                 | epay2.jpg     | JPG  | 8     | Social Studies           | pullf    | pay.mp3   |
| land1      | NA   | Land ownership                            | land1.jpg     | JPG  | 8     | Social Studies           | pullf    | lan.mp3   |
| land2      | NA   | Land ownership                            | land2.jpg     | JPG  | 8     | Social Studies           | pullf    | lan.mp3   |
| fami1      | NA   | Family                                    | fam1.jpg      | JPG  | 8     | Social Studies           | pullf    | fam.mp3   |
| fami2      | NA   | Family                                    | fam2.jpg      | JPG  | 8     | Social Studies           | pullf    | fam.mp3   |
| pull       | NA   | Pull factors of immigration               | imm1.jpg      | JPG  | 8     | Social Studies           | pullf    | pull.mp3  |
| free1      | NA   | Freedom                                   | free1.jpg     | JPG  | 8     | Social Studies           | pullf    | free.mp3  |
| free2      | NA   | Freedom                                   | free2.jpg     | JPG  | 8     | Social Studies           | pullf    | free.mp3  |
| pull2      | NA   | Pull factors of immigration               | imm2.jpg      | JPG  | 8     | Social Studies           | pullf    | pull.mp3  |
| free2      | NA   | Freedom                                   | free2.jpg     | JPG  | 8     | Social Studies           | pullf    | free.mp3  |
| ind1       | NA   | Industry needed workers                   | ind1.jpg      | JPG  | 8     | Social Studies           | pullf    | ind.mp3   |
| ice        | NA   | Invented machine to make ice              | 1844.gif      | GIF  | 8     | Technology               | irefrg   | 1844.mp3  |
| 1988       | 1988 | Internet relay chat                       | 88.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1988.mp3  |
| 1961       | 1961 | Packet switching theory                   | 61.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1961.mp3  |
| refr       | NA   | Invented refrigiration system             | 1805.gif      | GIF  | 8     | Technology               | irefrg   | 1805.mp3  |
| art        | NA   | Invented Artificial refrigiration system  | 1748.gif      | GIF  | 8     | Technology               | irefrg   | 1748.mp3  |
| tox        | NA   | Refrigerator contained toxic gases        | tgas1.gif     | GIF  | 8     | Technology               | irefrg   | toxic.mp3 |
| freo       | NA   | Freon replaced toxic gases                | freon.gif     | GIF  | 8     | Technology               | irefrg   | feron.mp3 |
| fre        | NA   | Freon replaced toxic gases                | fre1.jpg      | JPG  | 8     | Technology               | irefrg   | feron.mp3 |
| refr       | NA   | Invented refrigiration system             | 18051.jpg     | JPG  | 8     | Technology               | irefrg   | 1805.mp3  |
| icem       | NA   | Invented machine to make ice              | 18441.jpg     | JPG  | 8     | Technology               | irefrg   | 1844.mp3  |
| toxi       | NA   |  Early refrigirator contained toxic gases | tgas1.jpg     | JPG  | 8     | Technology               | irefrg   | toxic.mp3 |
| arti       | NA   | Invented Artificial refrigiration system  | 1748.jpg      | JPG  | 8     | Technology               | irefrg   | 1748.mp3  |
| friz       | NA   | Refrigerator                              | ref.jpg       | JPG  | 8     | Technology               | irefrg   | ref.mp3   |
| friz       | NA   | Refrigerator                              | refr.gif      | GIF  | 8     | Technology               | irefrg   | ref.mp3   |
| 1973       | 1973 | Gateway Architecture                      | 73.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1973.mp3  |
| 1988       | 1988 | Internet relay chat                       | 88.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1988.mp3  |
| 1964       | 1964 | Packet switching networks                 | 64.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1964.mp3  |
| 1966       | 1966 | ARPANET plan                              | 66.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1966.mp3  |
| 1971       | 1971 | Email program                             | 71.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1971.mp3  |
| 1962       | 1962 | Galatic Network concept                   | 62.gif        | GIF  | 8     | Internet  Timeline       | itline   | 1962.mp3  |
| 1966       | 1966 | ARPANET plan                              | 66.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1966.mp3  |
| 1971       | 1971 | Email program                             | 71.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1971.mp3  |
| 1973       | 1973 | Idea of Ethernet                          | 73.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1973.mp3  |
| 1974       | 1974 | Design of TCP                             | 74.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1974.mp3  |
| 1986       | 1986 | Mail exchanger                            | 86.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1986.mp3  |
| egy        | NA   | Energy is involved                        | egy.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | egy.mp3   |
| 1961       | 1961 | Packet switching theory                   | 61.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1961.mp3  |
| 1962       | 1962 | Galactic Network                          | 62.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1962.mp3  |
| 1964       | 1964 | Packet switching networks                 | 64.jpg        | JPG  | 8     | Internet  Timeline       | itline   | 1964.mp3  |
| rev        | NA   | Reversible                                | rev.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | rev.mp3   |
| nprp       | NA   | Properties do not change                  | nprp.jpg      | JPG  | 8     | Chemical-Physical Change | cpchange | nprp.mp3  |
| sta        | NA   | Change in state                           | sta.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | sta.mp3   |
| nrev       | NA   | Irreversible                              | nrev.jpg      | JPG  | 8     | Chemical-Physical Change | cpchange | nrev.mp3  |
| prp        | NA   | Properties change                         | prp.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | prp.mp3   |
| new        | NA   | New substances form                       | new.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | new.mp3   |
| all        | NA   | Happens to all matter                     | all.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | all.mp3   |
| rev        | NA   | Reversible                                | rev.gif       | GIF  | 8     | Chemical-Physical Change | cpchange | rev.mp3   |
| phy        | NA   | Physical change                           | phy.jpg       | JPG  | 8     | Chemical-Physical Change | cpchange | phy.mp3   |
| chem       | NA   | Chemical change                           | chem.jpg      | JPG  | 8     | Chemical-Physical Change | cpchange | chem.mp3  |
| adult1     | NA   | Complete Butterfly                        | cb1.gif       | GIF  | 8     | Biology                  | bcycle   | Adult.mp3 |
| adult2     | NA   | Complete Butterfly                        | cb2.gif       | GIF  | 8     | Biology                  | bcycle   | Adult.mp3 |
| fegg1      | NA   | Spawn (Egg-Mass)                          | egg1.jpg      | JPG  | 8     | Biology                  | fcycle   | f1.mp3    |
| fegg2      | NA   | Spawn (Egg-Mass)                          | egg2.jpg      | JPG  | 8     | Biology                  | fcycle   | f1.mp3    |
| tpole1     | NA   | Tadpole                                   | tpole1.jpg    | JPG  | 8     | Biology                  | fcycle   | f2.mp3    |
| tpole2     | NA   | Tadpole                                   | tpole2.jpg    | JPG  | 8     | Biology                  | fcycle   | f2.mp3    |
| ltpl1      | NA   | Tadpole with legs                         | ltp1.jpg      | JPG  | 8     | Biology                  | fcycle   | f3.mp3    |
| ltpl2      | NA   | Tadpole with legs                         | ltp2.jpg      | JPG  | 8     | Biology                  | fcycle   | f3.mp3    |
| flet1      | NA   | Froglet (Young Frog)                      | flet1.jpg     | JPG  | 8     | Biology                  | fcycle   | f4.mp3    |
| flet2      | NA   | Froglet (Young Frog)                      | flet2.jpg     | JPG  | 8     | Biology                  | fcycle   | f4.mp3    |
| cfrg1      | NA   | Complete Frog                             | frg1.jpg      | JPG  | 8     | Biology                  | fcycle   | f5.mp3    |
| cfrg2      | NA   | Complete Frog                             | frg2.jpg      | JPG  | 8     | Biology                  | fcycle   | f5.mp3    |
| fegg1      | NA   | Spawn (Egg-Mass)                          | egg1.gif      | GIF  | 8     | Biology                  | fcycle   | f1.mp3    |
| fegg2      | NA   | Spawn (Egg-Mass)                          | egg2.gif      | GIF  | 8     | Biology                  | fcycle   | f1.mp3    |
| tpole1     | NA   | Tadpole                                   | tpole1.gif    | GIF  | 8     | Biology                  | fcycle   | f2.mp3    |
| tpole2     | NA   | Tadpole                                   | tpole2.gif    | GIF  | 8     | Biology                  | fcycle   | f2.mp3    |
| ltpl1      | NA   | Tadpole with legs                         | ltp1.gif      | GIF  | 8     | Biology                  | fcycle   | f3.mp3    |
| ltpl2      | NA   | Tadpole with legs                         | ltp2.gif      | GIF  | 8     | Biology                  | fcycle   | f3.mp3    |
```

Figure 4.15 Imagedb Table

4.4 Multimedia Search Agent

Multimedia search agent is a critical component of vizipen system. The main functionality of this system is to search for multimedia elements on the web and store them in vizipen database. MSA is not fully functional in this thesis. According to the original design of vizipen MSA searches for the multimedia elements on the web and saves them in vizipen database, but we implemented only a portion of the functionality of this project. In this project MSA searches for multimedia elements in vizipen database based on the concept and multimedia element type selected by the end user. Vizipen database is pre populated with various multimedia elements collected from the

World Wide Web. As described in section 4.3 the database tables' imagedb and animdb contain multimedia elements that are already loaded manually.

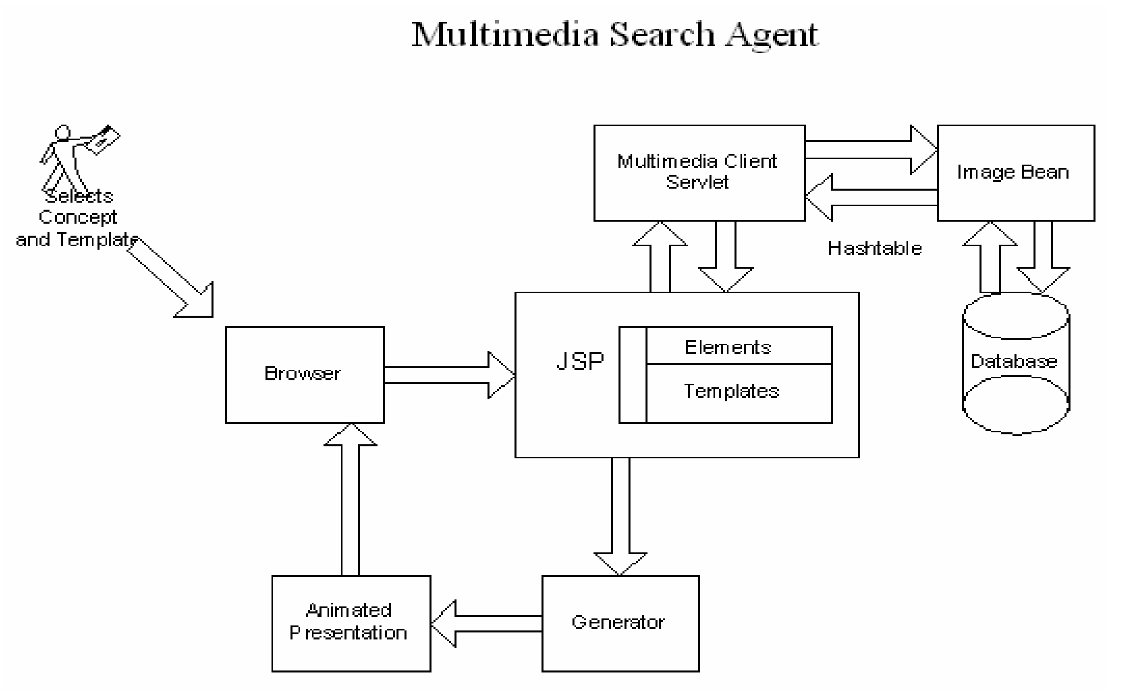


Figure 4.16 Multimedia Search Agent

As shown in figure 4.16, when the user selects a concept and multimedia element type, the request will be sent to multimedia client servlet, which sends a request to imagebean. Imagebean queries the multimedia database and sends a bean containing the multimedia elements in the form of a hash table to the multimedia servlet. The multimedia servlet sends these multimedia elements to the JSP page and the JSP page displays all these images in the web browser. The user can select, drag and drop the multimedia elements in the presentation template. The template and selected multimedia elements will be sent to the presentation generation engine, which constructs the animation by placing the multimedia elements in the placeholders. The animation generated by the presentation engine will be sent to the web browser, where it will be displayed to the end user.

4.5 Presentation Generation Engine

Presentation generation engine is the component that puts everything together and presents the final output to the end user. In this thesis we are using Jgenerator, which is a third party tool for generating animated presentation. A detailed description about Jgenerator was given in section 2.3. This engine (the “Generator” in the sequel) enables automated generation of presentations by replacing designated objects within the presentation or their properties. Generator is the server side software for automatically delivering dynamic, on-the-fly, personalized web graphics immediately or at scheduled times. The generator takes the animated template and inserts the content objects (text, image, animation, video etc) into the corresponding places in the template. The completed animation is then sent to the Web-Server to be delivered to the user.

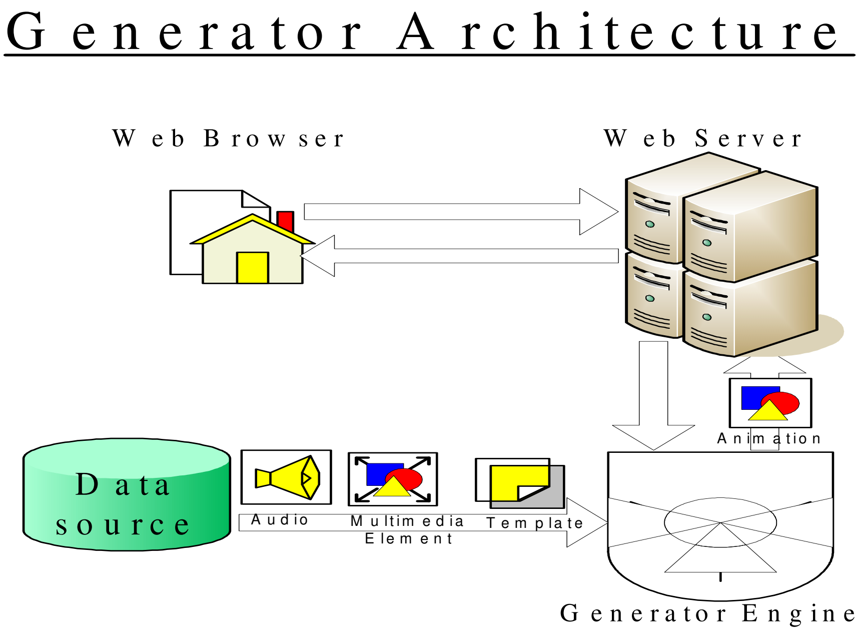
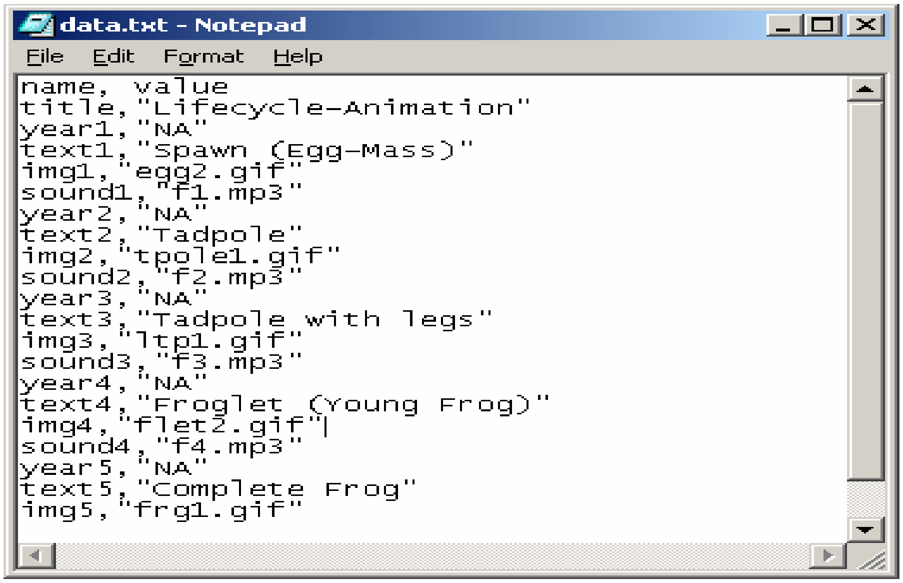


Figure 4.17 Generator Architecture



```
name, value
title,"Lifecycle-Animation"
year1,"NA"
text1,"Spawn (Egg-Mass)"
img1,"egg2.gif"
sound1,"f1.mp3"
year2,"NA"
text2,"Tadpole"
img2,"tpole1.gif"
sound2,"f2.mp3"
year3,"NA"
text3,"Tadpole with legs"
img3,"ltp1.gif"
sound3,"f3.mp3"
year4,"NA"
text4,"Froglet (Young Frog)"
img4,"flet2.gif"
sound4,"f4.mp3"
year5,"NA"
text5,"Complete Frog"
img5,"frg1.gif"
```

Figure 4.18 Generator Data Source in File Format

The presentation generation engine takes multimedia elements, data source specifying the values for the variables in the template and presentation template as the input and generates the presentation by placing the images and text data in the placeholders of the template. The final animation will be generated on the fly and presented in the graphical user interface, which is the web browser. As shown in figure 4.17, input for Jgenerator is provided from the user interface where the user drags and drops the multimedia elements into the presentation template. Generator takes multimedia elements, presentation template and data source (as shown in figure 4.18, a data.txt file prepared by the dtatadisplay component of the application) and combines all these elements to generate the presentation. Once the presentation is generated, it will be sent back to the user interface as an animation (.swf file).

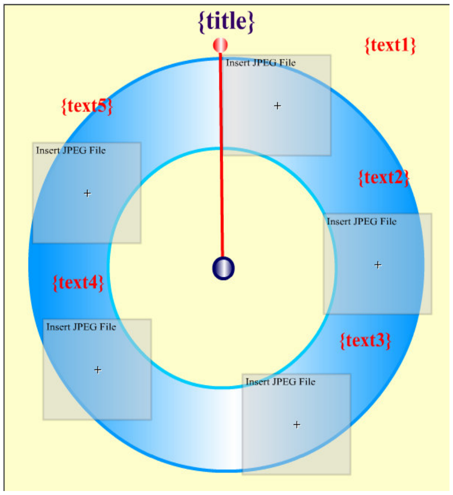


Figure 4.19 Presentation Template

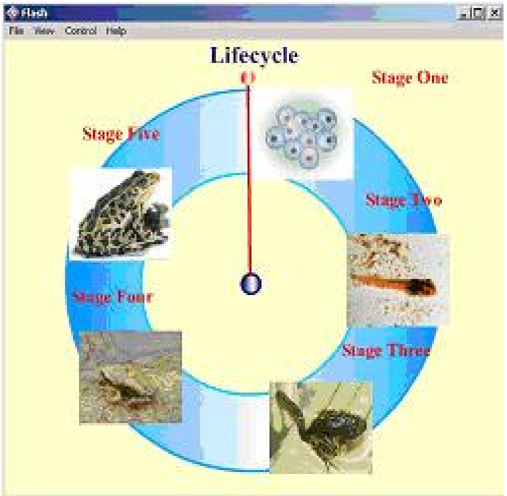
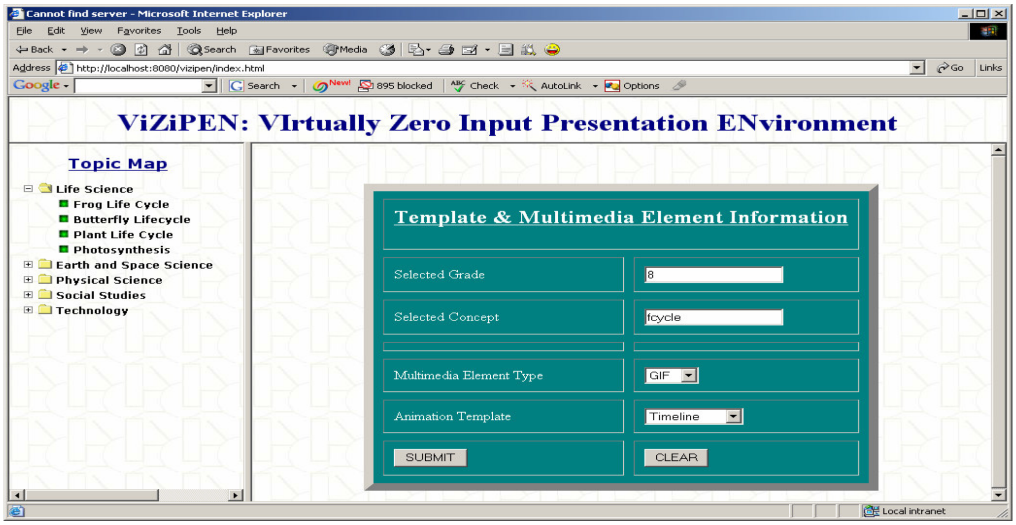


Figure 4.20 Animated Presentation

4.6 Graphical User Interface

The graphical user interface is also a critical part of the system. The end users interact with the system by using this. The GUI helps the user in selecting the concept, selecting the template, selecting multimedia elements, placing the multimedia elements in the placeholders and in viewing the animated presentation. All the GUI components are the web pages that are developed using HTML and JSP. HTML pages are used to show the static content in web pages where as JSP pages are used to show the dynamic contents. GUI component shown in figure 4.21 is the interface for selecting the concept, multimedia element type and presentation template. As shown in the figure topic map on the left most side of the screen shows the list of concepts for which we can generate the presentations.


Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/index.html
Go Links
Google Search New! 895 blocked Check AutoLink Options
SUBMIT
CLEAR
Local intranet

ViZiPEN: VIrtually Zero Input Presentation ENvironment

Topic Map

- Life Science
  - Frog Life Cycle
  - Butterfly Lifecycle
  - Plant Life Cycle
  - Photosynthesis
- Earth and Space Science
- Physical Science
- Social Studies
- Technology

Template & Multimedia Element Information

Selected Grade	8
Selected Concept	fcycle
Multimedia Element Type	GIF
Animation Template	Timeline

Figure 4.21 Template and Multimedia Element Selection

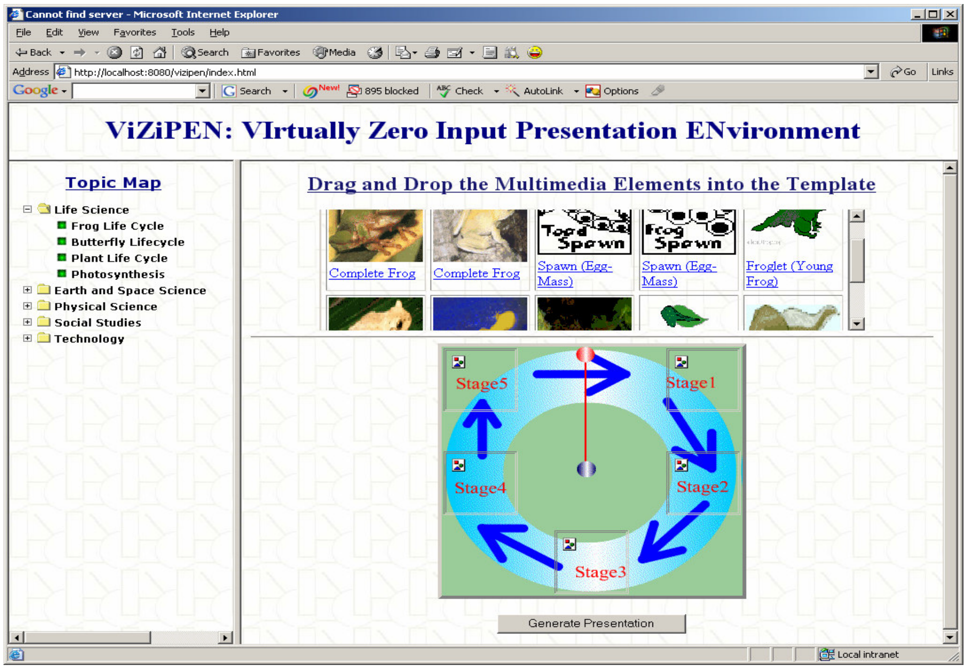

Address http://localhost:8080/vizipen/index.html

Figure 4.22 Multimedia Elements for Frog-Lifecycle Concept

The user can select the concept by clicking on the concept name and then select the multimedia element type and template type by selecting the drop down fields provided on the right side of the page. Once the selection is complete user hits on the submit button to go to the next level of the process. In the next level the user will be provided with a set of multimedia elements related to the concept selected and the presentation template where the user can place the images in the placeholders as shown in figure 4.22. When the user selects a concept and a template, a request will be sent to the Client Server and servlet queries the database for the multimedia elements and the presentation template. The result of the query will be sent back to the servlet, the servlet converts the multimedia elements and templates in the binary format (BLOB) to .jpg / .gif / .swf / .swt file formats and stores these files under a new directory on the disk and displays these files and template in the web UI. Figure 4.22 shows the multimedia elements in the upper half portion of the interface where as the lower half portion shows the presentation templates.

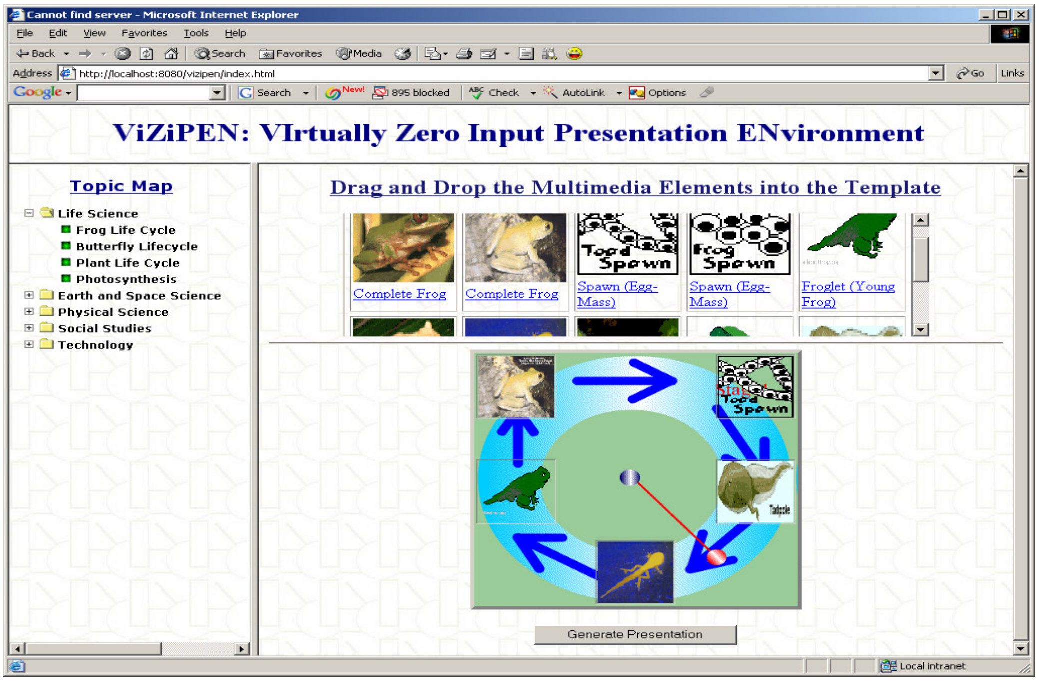

Cannot find server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/vizipen/index.html
Go Links
Google Search 895 blocked Check AutoLink Options
ViZiPEN: VIrtually Zero Input Presentation ENvironment
Topic Map
Life Science
Frog Life Cycle
Butterfly Lifecycle
Plant Life Cycle
Photosynthesis
Earth and Space Science
Physical Science
Social Studies
Technology
Drag and Drop the Multimedia Elements into the Template
Complete Frog
Complete Frog
Spawn (Egg-Mass)
Spawn (Egg-Mass)
Froglet (Young Frog)
Toad Spawn
Frog Spawn
Tadpole
Generate Presentation
Local intranet

Figure 4.23 Multimedia Elements Placed in Placeholders of Template

The user selects the multimedia elements and places them in the placeholders as shown in figure 4.23. Once the elements are placed in the placeholders' user clicks on the "Generate Presentation" button to generate the animated presentation. When the user places the elements in the presentation template, the names and properties of the images will be written to the data source file. The data source file is the input for the presentation; it provides the absolute path for the multimedia elements and the values for their properties. The presentation generation engine replaces the variables of the presentation templates and generates the presentation. The final presentation will be presented to the user as shown in figure 4.23.

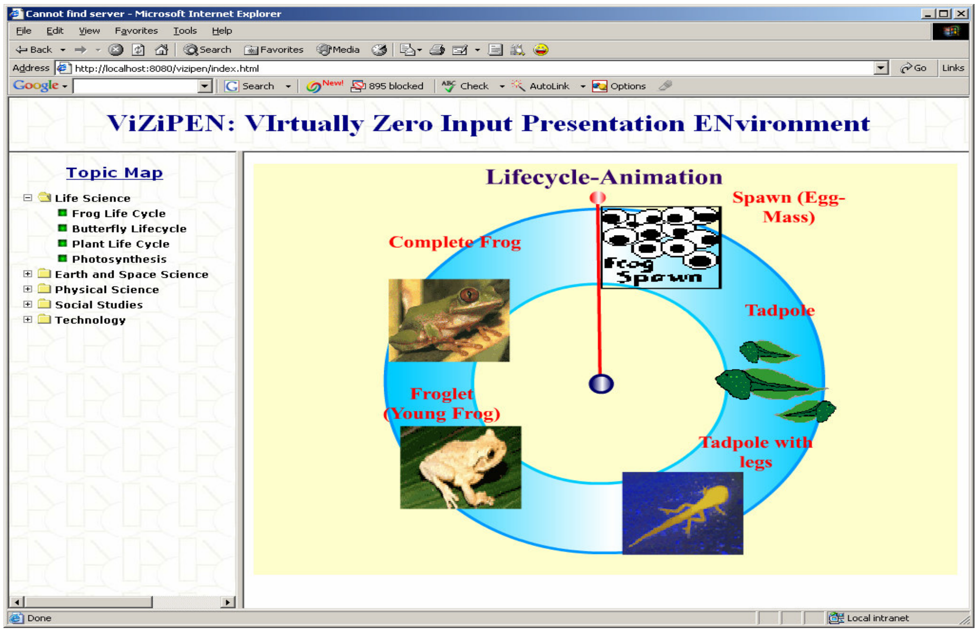


Figure 4.24 Frog Life-Cycle Animation

CHAPTER 5

RELATED WORK

The idea of constructing video presentations from the existing video segments under formal conforming to constraints is discussed by Hakkoymaz, Kraft and Ozsoyoglu (1999). A Java-based interactive system is described in Chen, Leng, Horn & Yang (2000). This system is used to teach geometrical concepts in mathematics. In Gregory & Kittler (2001), a technique for dynamic generation of customizable animations is described. Contextual information is retrieved from a knowledge source by using concept-based information retrieval techniques. Our use of the domain knowledge sources is similar to this approach. Semantics-based web searches have been discussed by Ozsoyoglu, Anderson and Ozsoyoglu (2000). The model described in that work uses Topic Maps as a guide to reach relevant pages, and Topic Meta-links to define navigational pathways on the web. Our approach for contextual topic representation is similar to this approach.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This chapter discusses about the conclusion, limitations and future work of Vizipen system. We have presented a system, named ViZiPEN, (VIrtually Zero Input Presentation ENvironment) for automatically generating multimedia presentations. The ViZiPEN system enables educators in general and K-12 teachers in particular to prepare animated visual presentations with minimal effort. ViZiPEN is designed to help teachers deal with the problem of lack of time in preparing educational materials for visual learning. Three key enabling technologies behind ViZiPEN are concept expansion, concept-based multimedia information extraction, and template based animation generation. We are currently working to improve our prototype system with additional content analyzers and XML-based representations of new domain knowledge sources in addition to school curriculums.

We are successful in associating multimedia elements with high school curriculum, which was the problem addressed by us. With our approach we are successful in generating dynamic animations using presentation templates and multimedia elements. We are also successful in illustrating concepts using interactive animated concept maps. Figure 4.25 shows the number of valid associations that we made from a set of sample concepts against a four presentation templates that we developed for this thesis. For this thesis we considered a total of 24 concepts and we collected multimedia elements for 16 concepts. Out of the 16 concepts we could represent 5 concepts using timeline, 4 concepts using lifecycle, 4 concepts using properties and 3 concepts using comparison template.

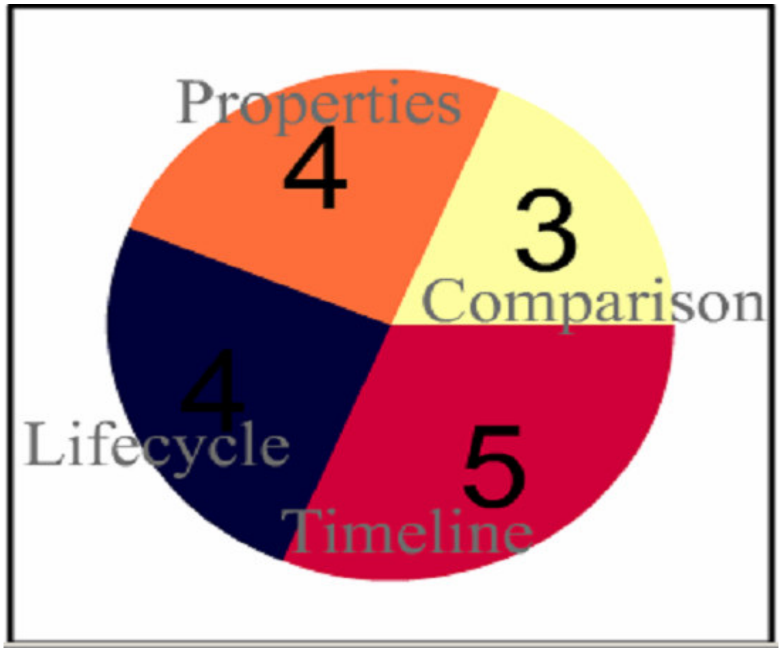


Figure 6.1 Sample Concepts to Template Mapping

Vizipen system also has some limitations, which will be considered as part of thee future work. Here follows the items, which will be part of future implementations.

- Automatic generation of Topic Map from curriculum: For this thesis we identified the topic map by following the curriculum. The topic map is being prepared manually. In the enhancement part, the topic map will be generated automatically.

- Multimedia Search Agent is not fully functional: Multimedia search agent is not implemented completely. We collected the multimedia elements and uploaded them to the database manually by using the user interface. In the enhancement work MSA collects the multimedia elements from the web and uploads them automatically to the database.

- Homogeneous templates: In this thesis we are supporting only homogeneous templates (i.e., each template accepts only one type of multimedia elements jpg or gif or swf etc). In the enhancement work we can develop templates, which can generate heterogeneous multimedia elements (one template will accept all types of multimedia elements).

- Limited Number of multimedia element types: In this thesis we are supporting only two types of multimedia elements JPG and GIF, in the future enhancement we can enhance the system to support more multimedia element types.

REFERENCES

[1] Gershon N., Eick S. G., and Card S., Information Visualization, ACM Interactions, vol. 5, no. 2, pp. 9-15, March/April 1998.

[2] Making Information More Accessible: A Survey of Information Visualization Applications and Techniques. By Gary Geisler, Last updated January 31, 1998.

[3] Card, S.K., Robertson, G.G. and Makinlay, J.D. The Information Visualizer: An information workspace. In: SIGCHI '91. Conference Proceedings on Human Factors in Computing Systems, 1991, pp. 181-188.

[4] D. Silver and X. Wang. Tracking and Visualizing Turbulent 3D Features. IEEE Transactions on Visualization and Computer Graphics, 3(2), June 1997.

[5] Bahari Belaton and Ken Brodlie. Model Centred Approach to Scientific Visualization, Journal of WSCG, Vol. 10, Number 1, pp. 63 -70.

[6] Kandogan, E. Star Coordinates: A Multidimensional Visualization Technique with Uniform Treatment of Dimensions. In Proceedings of the IEEE Information Visualization Symposium 2000, Late Breaking Hot Topics.

[7] Y. Alp Aslandogan, Suneel. Vana, P. Boppana, V. Mariappan, L.Stvan. "Empowering Educators in Visual Learning: Information Extraction and Visualization for Automatic Generation of Multimedia Presentations." Proceedings of SITE '03 (Society for Information Technology and Teacher Education), Albuquerque, NM.

[8] White Papers, Oracular, Vector vs. Bitmap Graphics: The differences and Pros and Cons of Each.

[9] M. Burge and W. Burger Learning Visual Ideals, in Proc. of the 9th ICIAP, (Florence, Italy), September 1997.

[10] White Paper, Macromedia - Developer Center: Macromedia Technology: Toward an Interactive Education Environment.

[11] Jgenerator: http://jzox.com/index.jsp

[12] Sun Microsystems, http://java.sun.com/products/jdk/1.2/docs/guide/jdbc/. January 2000.

[13] MySQL: http://www.mysql.com

[14] Fionnuala Flanagan. An Educational Enquiry into the use of Concept Mapping and Multimedia to Enhance the Understanding of Mathematics. ITTE Conference, Trinity College, July 2002.

[15] Aslandogan, Y.A., Yu, C. T. (2000). “Multiple Evidence Combination in Image Retrieval: Diogenes Searches for People on the Web”. Proceedings of the ACM SIGIR Conference, Athens, Greece.

[16] Miller, G.A., Beckwith, C., Felbaum, C., Gross, D. (1990). “Introduction to WordNet: An Online Lexical Database”. International Journal of Lexicography, 3:4, 235-244.

[17] Hakkoymaz, V., Kraft, J., Ozsoyoglu, G. (1999). “Constraint-based automation of multimedia presentation assembly”. Multimedia Systems Volume 7, Number 6, 1999, 7(6): 500-518 (1999).

[18] Chen, Y., Leng, T., Horng, A., Yang, D. (2000). “A Java-based Interactive Learning System of junior High School level Geometry”. Proceedings of International Conference on Computers in Education (ICCE/ICCAI2000), Taipei, Taiwan.

[19] Gregory, L., and Kittler, J. (2001). "Using Contextual Information for Image Retrieval". Proceedings of the 11 the Int'l Conference on Image Anaysis and Processing(ICIAP 2001). Palermo, Italy.

[20] Ozsoyoglu, G., Anderson, M.R., Ozsoyoglu, Z. M. (2000). "Web search with Metadata links and Multimedia presentations." Sixth Int. Workshop on Multimedia Information Systems. Oct. 26-28, 2000, Chica.

[21] Reichherzer, T. R., A. J. Canas, K. M. Ford, P. J. Hayes. The Giant: An Agent-based Approach to Knowledge Construction & Sharing, The Eleventh International Florida Artificial Intelligence Research Symposium Conference, Sanibel Island, (1998)

[22] Hong Chen. Towards Design Patterns for Dynamic Analytical Data Visualization, Proceedings of SPIE Conference on Visualization and Data.

[23] Yeong C. Kim, Martha W. Ewens, Joel A. Micheal and David Trace. Physiology tutorial using casual concept mapping, The 13th Midwest AI and Cognitive Science Conference, Chicago, IL-2002. PP 61-64.

[24] Jennifer Turns, Cynthia J. Atman*, Member, IEEE*, and Robin Adams. Concept Maps for Engineering Education: A Cognitively Motivated Tool Supporting Varied Assessment Functions. IEEE Transactions on education, Vol. 43, No. 2, May 2000.

[25] J. Fernando Vega-Riveros, Gloria Patricia Marciales-Vivas and Mauricio Martínez-Melo. Concept Maps in Engineering Education: A Case Study, Global J. of Engineering Education, Vol. 2, No. 1 Printed in Australia

[26] Akihiro Kashihara, Norio Matsui, Tsukasa Hirashima and Jun'ichi Toyoda. A Knowledge Visualization and Its Evaluation, Proceedings of ED-MEDIA '98, pp.679-684, Freiburg, Germany (1998).

[27] Akihiro Kashihara, Norio Matsui, Tsukasa Hirashima and Jun'ichi Toyoda. A History Visualization for Learning-by-Exploration in Hypermedia on WWW, Proc. of WebNet 98, pp.497-502, Orlando, FL, USA. (1998).

BIOGRAPHICAL INFORMATION

Suneel Vana received his Bachelor of Technology degree in Computer Science and Engineering at Kakatiya University, Warangal, India in 2001. He then pursued his Master of Science in Computer Science and Engineering at The University of Texas at Arlington. He is expected to receive his Master of Science in Computer Science and Engineering from The University of Texas at Arlington in December 2005. His research interests include multimedia database, bio informatics and software engineering. He is currently working as a Software Quality Assurance engineer at Lifesize communications, Austin, Texas.