2008

Design and Implementation of Online Learning Environments

Kaveh Jajouei-Moghaddam
Western University

Design and Implementation of
Online Learning Environments

(Thesis Format: Monograph)

by

Kaveh Jajouei-Moghaddam

Graduate Program in Computer Science

**A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science**

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Dr. Kamran Sedig	Dr. Bob Mercer
Supervisory Committee	Dr. Mike Katchabaw
	Dr. Immaculate Namukasa

The thesis by

Kaveh Jajouei-Moghaddam

entitled:

Design and implementation of
Online Learning Environments

is accepted in partial fulfillment of the
requirements for the degree of Master of Science

Date____________________

____
Chair of the Thesis Examination Board

Abstract

This thesis describes a systematic approach in the design and implementation of online learning environments. This approach incorporates the principles of human learning as well as the best practices in software engineering. This thesis implements a conceptual model for the design, and it describes how software elements can be developed to comply with the model. In the context of this research two online environments are developed and analyzed. The end product of this approach is a robust and reusable software architecture, a framework for design, and an effective and engaging model suited to online learning environments.

Keywords: representation, interaction, collaborative learning, mental model, epistemic activity, cognitive offloading.

Acknowledgements

To my wife, Leili, for her love and support which allowed me to achieve my goal.

To my parents, for their faith in my abilities and all their encouragement.

To my supervisor, Dr. Kamran Sedig, for his overwhelming support and guidance that brought this thesis to completion.

Table of Contents

List of Figures

Chapter 1
Introduction

Mathematics has been viewed by many students as a difficult subject. The feeling of resistance toward mathematics makes it a challenging area for students to engage with (Sedig, Rowhani 2008). As mathematics introduces a complex and yet interconnected body of concepts, dealing with mathematical domains requires a variety of skills and cognitive proficiencies. Learners of mathematics are often required to understand phenomena in terms of their quantities and interpret magnitudes that they try to represent. Moreover, learners are asked to properly connect the abstract or conceptual aspects of a mathematical idea with concrete instances of reality. Another aspect of mathematics deals with understanding structures and identifying inherent patterns. In this sense, learners are encouraged to discover the rules and the logic of a conceptual space, relate them to what they have previously contemplated, and generalize them. Computer-based methods of education are increasingly being adopted by teachers and educational institutes (Tapscott 1997, Urbanek 2001). Despite the promises that these types of software make, they often fail to deliver a desirable outcome in terms of improving students' performance and engaging them in mathematical content.

The goal of this thesis is to explore and develop a systematic strategy for designing and implementing online educational environments. The aim of these environments is to engage students in deeper learning of mathematics and stronger mathematical reasoning.

1.1 Statement of the problem

Web-based learning applications refer to a wide range of online software applications which are designed to support learning and teaching in an educational setting. In mathematics education, these software applications are focusing on mathematical contents. While some of these applications simply deliver mathematical contents in the form of online tutorials or hyperlinks, others tend to provide more interactive features and graphical representations. However, most designers of web-based applications which promote learning fail to base their design strategy on research evidences but merely on intuition (Gadanidis, Sedig &Liang, 2004). The inadequacy of incorporating the pedagogical concepts into design of online mathematical applications is evidenced by the huge amount of poorly designed software that is made available on the internet.

Extensive research in the field of human computer interaction and cognitive psychology indicates that learning as an internal process of the human mind is highly influenced by external visual representations (see Chapter 2). Moreover, active participation of learners in terms of interacting with visual representations enhances their overall understanding and thus improves their learning (see Chapter 2, and 3). Along with this line of thought, online interactive visualizations (IVs) such as those represented on the Illuminations web site of the National Council of Teachers of Mathematics (NCTM 2006) provide interactive mathematical objects. However, these objects are presented as fragmented pieces of mathematical topics, and they are passively scattered across several pages of a web site. Users pick one of the objects, interact with it, and form their own understanding. This approach primarily focuses on individual style of learning and completely disregards the significance of the social context in supporting the cognitive development.

Comprehensive research in both fields of knowledge management and collaborative learning emphasizes the influential role of knowledge sharing and social interaction on the cognitive performance of individuals (see Chapter 2). Thus, a great effort is required to establish a knowledge space in which the learner constructs his or her own knowledge and dynamically shares it with other learners.

On the other hand, the design and the delivery of most online mathematical components via the web are not maintained in such a way as to take advantage of the full potential that the technology of the World Wide Web can offer. For instance, in the design of Illumination website of NCTM, the web technology is merely utilized as a medium to deliver the interactive mathematical content. This view of design, fails to make the most of the web technology to create a dynamic and integrated environment, which embodies users' inputs, in order to dynamically enrich its content. In this sense, a pragmatic shift toward the designing of online mathematics learning is necessary to provide a liaison between pedagogical concepts and technological challenges ensuring the effective use of the technology.

This research attempts to investigate and explore the following areas in order to better understand and design online mathematical applications.

- Explore and identify the theoretical principles of human cognition in the context of individual and social settings in order to establish a pedagogical foundation supporting the design strategy.

- Translate the multi-dimensional and complex pedagogical principles into series of justifiable design strategies of software components.

- Promote the web as an integrated environment, as distinct from a partial delivery medium, which dynamically incorporates interactive visual components and facilitates social interaction among learners to encourage knowledge construction.

- Investigate the potential of current software architecture in identifying the strategies and establishing a foundation for supporting the pedagogical principles.

1.2 Approach

Along the lines of designing an enhanced online mathematical environment, fostering the key elements of research and practice, the following approaches are adopted:

In chapter 2, the current research develops a theoretical background in the areas of human cognition, representations, interactions, collaboration, and knowledge sharing. It articulates how human cognition evolves as a result of active participation in interacting with objects and individuals within an environment.

In chapter 3, I have developed a conceptual model which describes the structure of the integrated environment embracing the theoretical principles. Using the model, a collection of 13 mathematical topics was chosen to be explored. These topics are believed by teachers to be the primary topics of mathematics covering a wide range of objectives in the mathematical learning. For each topic, I have implemented a series of interactive visual components and deployed them on two web environments. More that 2000 students and teachers have been granted access to the environment and some of their comments are captured.

Moreover, chapter 3 details the pedagogical justification for the design of the interactive components. It also provides a description of usage for each of the interactive visual components.

Exploring the challenges of the design and implementation, chapter 4 describes the architectural rationales and requirements forming the design decision. It describes how designed modules are utilized to suit the pedagogical requirements.

Chapter 5 provides suggestions and illustrates the areas requiring more research for future works.

Chapter 2
Literature and background review

Well designed software is a product of solid understanding of the domain that it tries to represent (Kotonya, Sommerville 1998). In this sense, designers' understanding of the domain problem becomes a crucial factor (Rosson, Gold 1989; Henninger 1995; Blyth 1999; Adelson, Soloway 1985). Design of software, with the primary goal of improving learners' performances, cuts across a heterogeneous domain comprising several disciplines such as cognitive psychology, human computer interaction, multimedia study, and software engineering. Looking into the concepts introduced by these disciplines, this chapter provides a theoretical background to inspire the pedagogical planning and design strategies. It starts with illustrating the internal aspects of human learning by defining learning as the primary goal of the design, exploring knowledge as the key goal of the learning, and describing the mental modeling as the internal organization of knowledge. Then, it demonstrates how external aids such as visual representations and interactions, and the way they are designed, can enhance the learning. Lastly, it explores the impact of social contexts on the human learning.

2.1 Learning

Researchers consistently point out that most software products which promote learning suffer from lack of a robust foundation on learning theories (Baker 2000). As the primary goal of educational software, learning is a complex and multidimensional process, which leads to constructing knowledge. The challenge of understanding the nature of learning comes from the integrating of the enigmatic concept of knowledge with the complex system of human mind. Scientists have presented different views of learning. The theoretical principles, introduced by each view, recommend a specific style of learning, and so impose a distinctive design strategy for software.

Biologists suggest that all living organisms have a capacity for obtaining and processing environmental feedback. The ability to learn from the environment is the basic skill to assure the survival of organisms (Aring, Rhem & Liljenström 1997). It is an attribute, which makes the organisms more durable and stronger in terms of dealing with the environment that is surrounding them.

Behaviorist psychologists have attempted to explain learning as the natural reaction of individuals to external phenomena. Formation of a behavior is described as what humans learn from the environment. This perspective suggests that learning can further be reinforced through reward or punishment (Alessi, Stanley & Alessi 2001).

Theories such as Gestalts describe the learning as a need or desire to maintain mental equilibrium. Acquiring consciousness about physical and environmental phenomena can satisfy mental equilibrium. It also relates to humans' internal perceptions and understanding of external phenomena. Facts which conflict with humans' internal understanding will create a level of negative feeling which pushes them towards readjusting their mind to regain equilibrium. Edgar Schein (1999) describes this motivation, which leads humans toward learning, as "survival anxiety". It also explains why under difficult circumstances individuals learn more efficiently. In a similar approach, Ackoff (1999) uses the analogy of driving under stressful and unanticipated situations to show how one's learning is amplified.

Cognitive science, with its primary focus on mind or intelligence (Luger 1994), has incorporated relevant disciplines including psychology, neuroscience, biology, and computer science, seeking to understand the nature of human cognition. Cognitive science introduces the notion of cognition as the ability to learn. It describes the cognition as the product of creating association among nodes of concepts, ideas, and memories. The path through the "vast network of relationship" from actual "percepts" to "abstract concepts" provides meaning and identity for those concepts (Sowa 1984, p76). Sowa (1984) also characterizes the nature of a concept as the identity of that concept which eventually links to materials that are evidently applicable and can be perceived through the senses.

This interpretation of cognition resembles the analogy of a graph with connected nodes. The interconnected graph of associations in which concepts (vertices) and semantics (edges) represent knowledge is the fundamental of a hypothesis called semantic network (Luger 1994). A semantic network represents a dynamic and evolutionary model for the development of knowledge within the human mind.

While all perspective of learning asserts on the function of environment, the cognitive outlook of learning appears to regard the mind as an active agent interacting with the environment. Cognitive science introduces the notion of epistemic actions as external actions that the mind performs to make sense of the external phenomena (Chandrasekharan, Stewart 2007; Kirsh, Maglio 1994). Epistemic actions support the internal computation of the mind and portray a clearer and more understandable picture of the phenomena (Maglio, Wenger & Copeland 2008). For instance, to predict the end result of the rotation of a 2D shape, one may record the intermediate stages of this process to offload some of the computations which would be performed by the memory otherwise.

Epistemic actions and their role in making sense of external environments, as viewed by the cognitive perspective of learning, provide a relevant application in the design of computer-supported learning environments. That is, the effectiveness of a computer-based learning environment depends on its potential to support the mind through performing epistemic actions.

As mathematics introduces a complex and abstract structure, supporting the learners through performing epistemic actions and cognitive activities becomes imperative. Moreover, it is crucial to identify types of cognitive actions that are required in mathematical learning. The following sections explore some of the common cognitive activities which are required in understating mathematical concepts. These activities are listed as working memory, searching, detecting salient patterns, reasoning, and interpreting.

2.1.1 Memory

Memory as a psychological faculty stores the path through the network of semantics and provides a mechanism for retrieving information and concepts. In addition, it dynamically changes as the network itself evolves. Some of the links are reinforced as the result of being referred to or reapplied, which explains why rehearsing improves the memorizing of information.

Memory is often considered as a combination of both short term and long term storage (Atkinson, Shiffrin 1968; Gazzaniga 1988; Cowan 2005). Short term or working memory allows us to keep information over a short period of time, while long term memory is utilized to store them in longer periods, possibly, around decades. In mathematical topics, short term memory is frequently utilized to keep track of intermediary stages of calculation, visual relations, transformations, and so on (Vandervert 2003).

Furthermore, working memory plays a key role in performing higher cognitive activities such as reasoning and sense making (Colom et al. 2004; Kyllonen, Christal 1990). However, working memory is observed to be limited in its performances (Kirschner 2002). Functions of working memory can be carried out, or to some extent, replaced by computers.

2.1.2 Pattern discovery

Mathematics is described as the science of discovering patterns and understanding the relations (Devlin 2003; Steen 1988). In mathematics, pattern discovery is involved with the process of identifying and relating the relations within a structure. Through this process, the mind is actively engaged with the basic epistemic action of mental scanning, searching, and comparison. Mathematical software can exploit the visualization techniques to support learners through identifying and recognizing the inherent patterns within a structure. These techniques will be further explored in this research.

2.1.3 Interpretation

Interpretation is the process of making sense of things and assigning a meaning to them. Brey (2005) describes the interpretation of perceptual data as a way by which "Perceptual stimuli are made useful as objects of conceptual thought, which does not range over sensory images but over concepts". This view implies that interpretation is a translation of data from perceptual space into a collection of meaningful concepts. Interpretation can be achieved either quantitatively through measuring objects and percepts or qualitatively by categorizing and assigning the percepts with concepts. In mathematics, it is very common to make sense of a phenomenon by interpreting the graph representing that phenomenon. Computer-based visualizations can provide visual cues easing the process of interpretation.

2.1.4 Inference and reasoning

Inference and reasoning are high-level cognitive activities that are very common in the practice of learning mathematics (Sherry 2006). Inference can be thought of as ways that humans derive conclusions out of what is already known. Inference may occur through inductive or deductive reasoning. Inductive reasoning refers to reaching a conclusion based on set of evidence or premises (Foltz, Overton 1995). In other words, it is the act of generalization. Conversely, deductive reasoning is identifying a particular truth from general evidence. Well designed software simulators can support learners to infer logical conclusions by providing them proper and relevant evidence (Chang et al. 2008; De Jong, Van Joolingen 1998).

2.2 Mental Model

A mental model is described as an internal model representing external phenomena.

This model is the product of human cognition mapping the external world internally.

In general, mental models are tightly related to the way humans understand the world and make sense of that internally. It is also linked to the way they make up their mind, organize their thoughts, specify the rules, generalize the facts, ignore trivial points, discover patterns, anticipate a course of action, and make decisions (Coll, Treagust 2003). Mental models are the end result of learning, and they appear to be the primary driver for human reasoning. In mathematics, learners use their mental model, or prior knowledge, to reason and solve mathematical problems (Chinnappan 1998).

Relevantly, it is fundamental for designers of learning environments to have a clear understanding of the nature of mental models. Researchers assert that software systems should provide features which are well adapted to their user's mental model (Rogers, Rutherford & Bibby 1992; Norman 1990). In this sense, the effectiveness of a learning software system is highly correlated to the extent to which the system appropriately influences its learners' mental models.

Experiments in both the fields of cognitive neuroscience and cognitive psychology postulate the existence of mental models and their functions (Johnson 1990; Kosslyn 1994). However, the nature of mental models is still the topic of debate. The term "mental model" was first coined out by a Scottish Psychologist, Kenneth Craik (1952). He refers to mental model as "small scales models" of reality that we use to explain phenomenon and anticipate events. In this respect, mental models are akin to architecture, maps, and diagrams in that their structures correspond to what externally exists, in contrast to what is logically formed.

Similarly, Norman (Norman 1983) describes the formation of a mental model as the result of "interacting with the environment, with others, and with the artifacts of technology". He further asserts on the power of this model as an "explanatory" for "understanding of the interaction".

Johnson-Larid (1983, p397) presents a different perspective of mental model. He emphasizes the role of mental models as the fundamental structures for human cognition. He describes the "central and unifying role" of mental models in "representing objects, states of affairs, sequences of events, the way world is, and the social and psychological actions of daily life". He also believes that the source of a mental model is perception.

The process of creating such a map is often termed cognitive mapping, which ultimately generates a cognitive map (Ruddle et al. 2000; Spence 1999; Spence 2001). In this context, a map is associated with cognition and is not simply a replica of the world. Thus, it is more representative of the active and dynamic interpretation of mental model.

Mental models in general have the following characteristics (Doyle, Ford 1998).

- They are subject to constant changes through a self-correcting mechanism.
- They are not accurate representations of reality.
- They contain minimal elements of reality, just enough to interpret the reality.
- They are situated and the level of their accuracy depends on the context and situation in which they are built.
- They are rule-based maps.

In short, mental models depict the internal aspects of learning. They represent what and how ideas and knowledge are perceived within the human mind. Following sections explore how representations, externally, contribute to the development of mental models.

2.3 Representations

The Merriam-Webster English Dictionary defines "representation" primarily in respect to presenting objects, facts, and ideas. Peterson (1996) and Hermian (1996) similarly define representations as notations or formats which can be modified and interpreted. Within these definitions, there is an implied visual component that is the end product of the encoding, transforming, and reconstructing of an original phenomenon. In other words, things are articulated differently from their initial form. The product of this articulation can be either externally expressed or internally manifested.

Internal representations are interchangeably used as mental models. Construction of a mental model is not exclusively an internal process. We are highly influenced by the environment that surrounds us. The environment provides us with basic perceptual inputs such as objects, shapes, and sounds that, later on, form our mental models. Our high-level cognitive activities such as sense making, reasoning, and discovering are also influenced by the environment. In line with this view, Simon (1996, p6) uses the similar term of inner and outer environment.

Therefore, to design a learning environment, it is necessary to consider internal and external representation as a unified representational system or a "distributed cognition" (Zhang, Norman 1994; Norman 1988; Hutchins 1990). The following section elaborates the external visual representations (VRs) and how they facilitate human cognition. Exploring the external VRs is aimed toward developing an effective strategy in the design of visual mathematical software components.

2.3.1 External Visual Representations

External visual representations (VRs) are encoded versions of objects, reality, relations, semantics, functions, and properties presented as a set of graphical symbols (Tufte 2000).

VRs support the transferring of knowledge and amplify the cognition (Norman 1993). They provide visual clues to conceptual structures (Ware 2000). VRs render the abstract complexity of objects, concepts, and ideas and translate them into more visually perceivable structures. In relation to the design of a learning environment, it is imperative to be clear on how external visual representations can be utilized to support learners through reasoning, mental modeling, and other epistemic actions. In this sense, the quality of a learning environment appears to be highly related to the way that it visually represents educational materials. Complying with this line of thought, following sections detail how external visual representations can aid learners and how they influence their mental processing.

2.3.2 Visual representations extend cognitive resources

VRs enhance cognition by extending cognitive resources. Norman (1993) asserts that VRs extend the working memory. They offload cognitive effort by grouping informational elements and constructing logical relations (Larkin, Simon 1987).

VRs are able to supplement human cognition by presenting some features, which to some extent are achievable, by the mind. However, it takes too much time and effort to identify those. Examples of these features are color-coded paths through a map. Overall, VRs improve the action of the human mind in keeping and storing information.

Nevertheless, VRs do not replace high-level cognitive processes such as sense making and reasoning. For the most part, they improve the performances of those processes by allocating visual resources, helping to offload low-level cognitive tasks (e.g., memorizing, searching).

2.3.3 External VRs assist interpretation

In addition to extending our cognitive recourse, visual representations can influence our mental model by enforcing structures or imposing relations (Qin, Simon 1992). Consequently, they facilitate interpretation. Interpretation can be either quantitative or qualitative (Brey 2005). Qualitative interpretations are mostly concerned with assigning a size to an object or get a sense of its measure. For instance the representation provided in the Figure 2-1 enforces the relation between different statues by comparing their sizes.

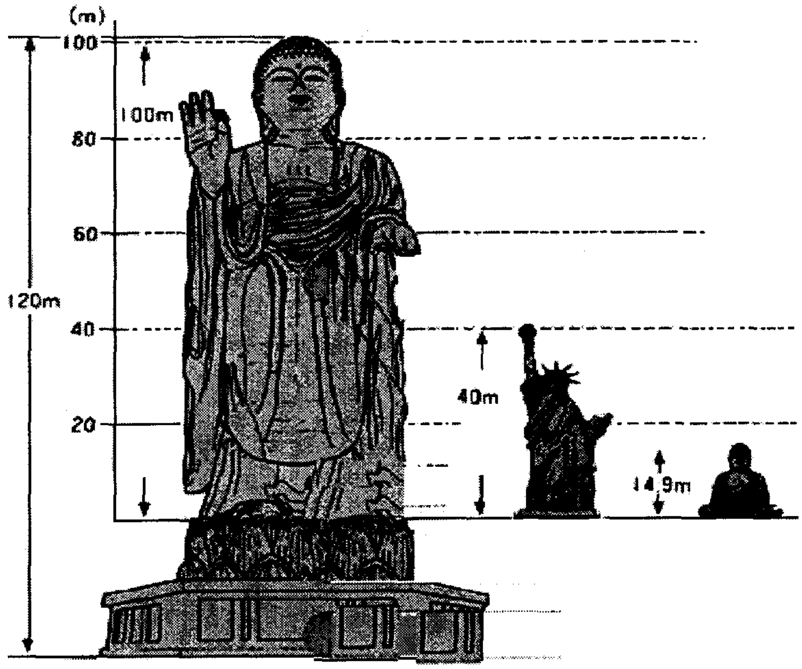

Figure 2-1 Quantitative representations, adapted from http://umi2tsukuba.files.wordpress.com/2008/06/cimg3765.jpg

On the other hand, qualitative interpretations are mostly concerned with assigning "categories" or "concepts" to a set of data (Brey 2005). For instance the VR illustrated in Figure 2-2 enforces the relation among animals by putting them into categories of mammal, birds, and etc. Another example is a map of landscapes in which different colors are used to show different categories of lands.

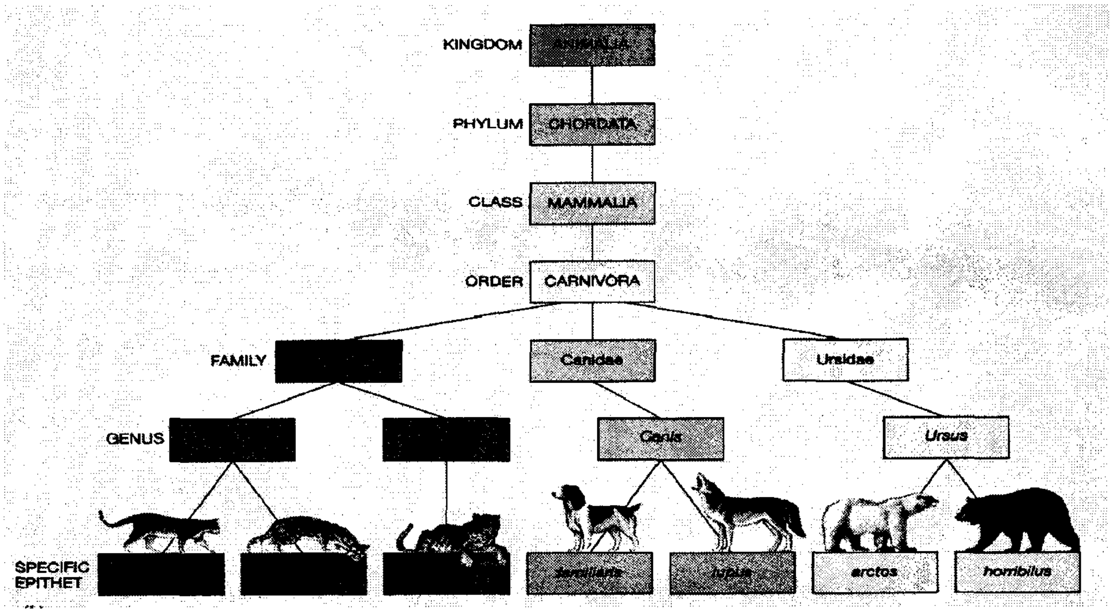

Figure 2-2 Qualitative representations, adapted from http://mac122.icu.ac.jp/gen-ed/classif-gifs/animal-class-example.gif

2.3.4 External VRs allow relevant points to stand out

Real world phenomena usually expose learners to enormous amount of information, which are irrelevant to the key concepts that they want to focus or explore. This information mostly disrupts the process of learning as it consumes the cognitive resources. In those cases, VRs can filter irrelevant information in order to highlight relevant points (Ware 2000). As such, salient structures and patterns are brought together to allow users to learn intended concepts and build hypothesis (Norman 1993; Sedig et al. 2003). This type of representations introduces the concept of the fidelity to the real world phenomena. Fidelity is defined as the degree to which a representation looks like the real world phenomena (Alessi, Stanley & Alessi 2001). On the other hand, an over simplification of phenomena may lead to a naïve understanding of the concept (Keith 2004). In this sense, a gradual movement from low fidelity to high fidelity is more desirable.

2.3.5 External VRs improve learning of abstract concepts

On the other hand, many conceptual representations are complex with an intricate interior embedding a high level of abstraction (Norman 1991; Skemp 1979). In these cases, visual representations can represent different level of abstraction to gradually shift the mind from fully abstracted concepts to more concrete objects creating a sense of relevance. Thus, VRs can provide a visual cue to the world of non-visual and abstract concepts.

2.3.6 External VRs aid Spatial and temporal reasoning

VRs facilitate a shift from analyzing data space to spatial and visual cognition. This can reduce the mental effort for understanding a complex body of information by taking advantage of humans' innate power of spatial reasoning. For instance, looking at a UML state diagram that forms links between different states of a process, is easier to understand than reading the procedural steps of the algorithms and functions, which detail the same idea. In addition, VRs can help to capture snapshots of temporal phenomena and represent them in spatial maps allowing a user to understand the transitional process (Sedig, Rowhani & Liang 2005).

2.3.7 Design of external visual representations

Norman (1993) brings up the notion of represented world and representing world. In his view, representing world is a set of symbols and relationships, which encodes the represented world or the represented reality. However, the way that the representing worlds visually encode the represented world can significantly change the quality of our learning. It can decide what information, structure, concept, or process should be explored (Zhang 1997). Norman and (Zhang, Norman 1994) define the term "representational effect" to address the same concept.

The following sections describe how different representations and the way they are designed can have different impacts on the learning.

2.3.8 Design of Depictive Vs Descriptive Representations

Represented world can be encoded as a set of depictive or descriptive symbols (Peterson 1996, Schnotz 2002). While both are using visual components to convey their meaning, their effects are different in terms of mental modeling. They also may represent inferably equivalent information (Larkin, Simon 1987) but the demand they put on the cognitive process of the observer are different. For instance, it takes more effort to read a direction than to simply look through a map as shown in Figure 2-3.

Figure 2-3 Depictive versus descriptive representations in Google map

Part of the reason that descriptive representations demand more cognitive effort is because they are discrete and disconnected in nature. Users have to first discover or construct semantic relations in order to make sense of them. In contrast, depictive representations come with built in relations. The innate power of the human mind in spatial cognitions and its constant reinforcement through interacting with the environment is another reason, which makes understanding of the depictive representations easier.

2.3.9 Design of additive and substitutive external VRs

Norman (1991, 1993) identifies two modes of representations as follows:

- Additive representations
- Substitutive representations

In substitutive mode, replacing one another carries out changes in the state of representation. Each representation appears in a stand-alone form and does not provide any visual cue to its relation with the previous or next representation. Hence, the responsibility to cognitively process and make sense of those relations is conveyed to the learners.

On the other hand, in additive mode each successive representation contains the previous one. Offloading some of the cognitive process, such as searching and tracing, makes it easier for learners to make comparisons and capture the inherent relations among represented elements. Examples of additive and substitutive representations are depicted in Figure 2-4.

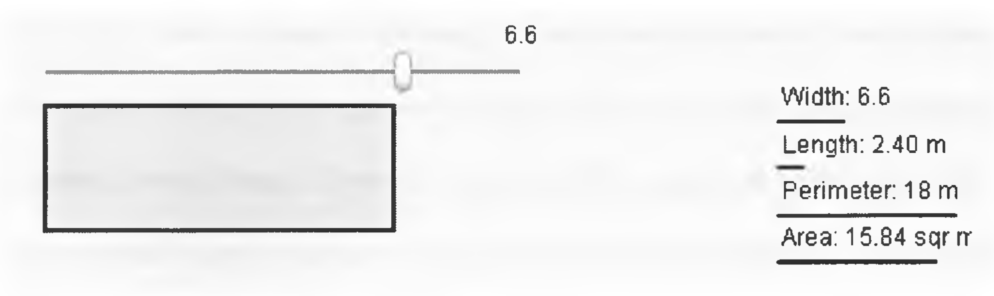

Figure 2-4 Additive versus substitutive representations

In the above figure, users can drag the slider to change the size of the rectangle. The size of rectangle constantly changes and the new values replace the old ones. In this case, numbers forms a substitutive representation.

Additionally, under each value there is a line, the length of which corresponds to the represented value. As the line shrinks or expands, its length provides a visual cue indicating the scale of the changes. In this case, lines depict additive form of representations.

2.3.10 Design of external VRs to promote deep learning

There are situations such as those described above where the goal is to put less cognitive demand on users. Therefore, representing a model that mostly corresponds with what users are already familiar with is a reasonable solution. Such solutions often take advantage of a concept called intuitiveness that means "known or perceived by intuition" (Merriam Webster English Dictionary). Intuition can be loosely inferred as a mental model which most people share. Thus intuitive representations are those representations that are in harmony with most people's understanding and hence, demand less cognitive effort. These intuitive representations reduce the distance between a user's intuition and represented world and they offer a "feeling of directness" (Hutchins, Hollan & Norman 1987).

However, when it comes to learning, it is not always desirable to reduce the cognitive effort, particularly when high-level cognitions such as sense making and reasoning are required. Sedig (Sedig, Klawe & Westrom 2001) reports that the quality of learning is not necessarily proportional to the intuitiveness of representation. They showed, in some cases, intuitiveness might also produce poorer and shallower learning results.

In addition to reducing cognitive effort, intuitive representations take on less user attention or engagement. Attention plays a significant role in the quality of learning (Sedig, Klawe & Westrom 2001; Ormrod 2007), as "it determines whether incoming sensory information is lost or processed" (Sedig, Klawe & Westrom 2001).

Moreover, researchers have shown that multiple representations of a phenomenon may also produce a deeper understanding of that phenomenon (Ainsworth 1999). Multiple representations promote abstraction and support generalization.

2.3.11 Design of external VRs to promote reasoning

The way we rationalize a situation will strongly influence our decisions and the way we ultimately perform a task. Different types of VRs can encourage different types of reasoning. In a study conducted by (Parnafes, Disessa 2004), they have shown how different but isomorphic representations of a particular phenomenon (in this case motion and acceleration) have promoted different inference and problem solving strategies.

This conclusion also implies that the design of representations should be in compliance with objectives of the tasks and the strategies that are required to perform the task.

In summary, since VRs play a significant role in constructing our mental models and primarily influence our knowledge and our views about concepts or phenomena, the way those concepts or phenomena are represented to us becomes considerably crucial. Thus, designers should systematically analyze and identify what, when, where, and how to represent the VRS to efficiently enhance the process of learning.

2.3.12 Static visual representations

External visual representations encode components of the real world in either descriptive or depictive form to convey the organization of ideas, logics, and concepts. In this sense, they play the role of communicative medium, facilitating the transfer of those concepts. However, external VRs are restricted in their communication when they are presented statically (Sedig et al. 2003; Sedig, Liang 2007). The next sections illustrate how interactions augment the communicative power of static visual representations.

2.4 Design of interaction

Interaction is the process in which a user acts upon a visual representation via the user interface. Sedig and Liang (2006) describe the role of interaction in terms of epistemic and communicative extensions to static visual representations. Following this point of view, interaction can be seen as an intermediary channel between information and mind (Sedig, Liang 2006). Interactions, also allow users to access information in an on demand fashion (Ware 2000).

From the design perspective, user interaction can be implemented to be performed through manipulation (Gadanidis, Sedig & Liang 2004). Manipulation can be either direct or indirect. In direct manipulation, users directly interact with the represented visual objects. An example of direct manipulation is when users drag a 3D object to rotate it. Indirect manipulation is performed when users use intermediary points to interact with the visual objects. An indirect manipulation for the previous example is when users use a slider to indirectly perform the rotation. The feel and cognitive effect of these two manipulation modes are studied to be different and have different impact on the learners' quality of learning (Sedig, Klawe & Westrom 2001;

Gadanidis, Sedig & Liang 2004; Nanda, Liang & Sedig 2005). From the pedagogical perspective, the direct and the indirect manipulation are loosely termed object and concept manipulation respectively (Sedig, Rowhani & Liang 2005). In that context, users either manipulate a representational object directly, or operate on the underlying conceptual elements of that object indirectly. For example, moving of a visual object on a grid can be performed either by using mouse to drag the object directly or by manipulating its coordination indirectly. In this case, the coordination is the underlying concept. The outcomes of the learning for the object manipulation and the concept manipulation are observed to be different (Sedig, Rowhani & Liang 2005).

In addition, the level of users' flexibility in manipulating the representations as well as the degree of reversibility of the system which allows users to correct their mistakes introduces a significant consideration in the design of the interactions. To some extent, a flexible and reversible interaction promotes an investigative learning style (Gadanidis, Sedig & Liang 2004; Shneiderman, Wald 1987) by allowing users to perform the trials and errors. However, improvised flexibility of manipulation may expose users to superfluous and irrelevant contents.

Sedig and Sumner (2006) have developed a general taxonomy of the low-level interactions. These interactions are intended to provide design options for developers. A summary of these interactions is listed in the Figure 2-5.

Interactions	Descriptions
Animating	Provide movement within VRs.
Annotating	Place note on VRs.
Chunking	Create relation among elements of VRs by grouping them together.
Composing	Constructing VRs using visual elements.
Cutting	Removing unsolicited elements.
Filtering	Hiding or showing part of VRs.
Fragmenting	Breaking VRs into pieces.
Probing	Focus and drill into some aspects of VRs.
Rearranging	Changing the position of elements in VRs
Re-picturing	Providing alternative representation of VRs
Scoping	Provide insight into the visual construction/deconstruction of VRs
Searching	Looking for features, objects, or relation within VRs

Figure 2-5 taxonomy of low level interactions

2.4.1 Feedback

One of the most important aspects of the design of the interaction is embedding appropriate feedback. Feedback allows users to visually perceive the outcomes of their actions and support them to reflectively monitor their own cognitive development and form hypotheses (Alessi, Stanley & Alessi 2001a). Feedback can be designed as immediate feedback or late feedback (Corbett, Anderson & Anderson 2001). While immediate feedback instantaneously represents the outcomes of the action, the late feedback introduce a delay in representing them.

Immediate feedback supports the exploration and investigation of a phenomenon by allowing users to act upon the representations and observe the effects of their actions (Gadanidis, Sedig & Liang 2004). On the other hand, late feedbacks support users through contemplating the potential outcomes and making decision.

2.4.2 Benefit of interaction

The above list of interactions can support users to understand the inherent concepts and dig into the logic and organization of represented VRs. They also allow users to investigate the relationships among various representations. Relevantly, they provide a foundation for the design of interactive visualizations.

Scholars stress the benefit of interaction as a mean to enhance a learner's cognitive process (Kirsh, Maglio 1994; Sedig, Liang 2007, 2006; Gadanidis, Sedig & Liang 2004; Kirsh 1997; Wagener, Strothotte 1998). The following list summarizes the general benefit of interactive VRs.

- Facilitate exploring, inquiring, navigating, and transforming VRs.
- Promote emergent understanding of encoded ideas (Sedig, Rowhani & Liang 2005).
- Synchronize internal cognitive models with the external VRs allowing users to form a more precise understanding of concepts.
- Offers more choices and a higher level of freedom for users to perform a task.

- Support temporal phenomena as well as spatial phenomena.

- Grant visibility to user's action by providing feedbacks (Norman 1988; Kirsh 1997).

- Facilitate situated learning.

- Encourage users' engagement to actively pursue a solution or perform a task.

- Bring the VRs closer to the users by making VRs "more responsive to the cognitive needs of users" (Kirsh 1997).

- Propose environments that are aligned with what humans in general do to perform a task or explore a phenomenon.

- Facilitate the high level cognition such as problem solving, decision-making, and mental modeling (Kirsh, Maglio 1994).

2.4.3 Design of appropriate interaction

Although most scholars agree on the significant role of interaction in promoting knowledge acquisition and development of cognitive skills (Barker 1994), the level of interactions and extents in which interaction should be applied is the topic of an ongoing discussion.

The interactivity in general is considered to be improved, if the level of the control and the degree of users' actions are extended. However, this view disregards the fact that some interactions may present ineffective or even disruptive cognitive loads. These ineffective loads are termed "extraneous loads", which are imposed by activities that do not contribute to the process of knowledge acquisition (Paas et al. 2003).

From another point of view, Spector (1995, p.531), while asserts on mental engagement and involvement as the most significant factors, recommends the creation of "more conversational interfaces" as a way of enhancing the interactions.

Sedig (Sedig, Liang 2007) represents the appropriateness of interaction in terms of "five different, yet interrelated, dimensions of fitness": *semantic*, *task*, *ontology*, *learner*, and *context*. These criteria propose a strategy toward the designing of interaction.

Semantic fitness

Semantic fitness is the quality of interaction to effectively convey the meaning and intended concept that a VR tries to communicate (Sedig, Liang 2007). Thus, interaction is considered to be appropriate if it can direct learners toward understanding and discovering the underlying concept of VRs.

In this sense, the dynamic linking among representations as well as the flow of interaction can be important to reveal the underlying semantic. Dynamic linkage refers to the defined direction and connection among representation (Gadanidis, Sedig & Liang 2004). These connections can be defined as singleton, bidirectional, or in the form of a network of representations. In network form, interacting with any of the representations will cause all other representations to adjust their forms to the current state.

Task fitness

Task fitness is the quality of VRs to facilitate performing tasks in directions that allow learners to acquire the intended skill or knowledge (Sedig, Liang 2007). As tasks are considered to be goal oriented activities, interactions should be designed in a way that thoughtfully directs users toward achieving those goals.

Ontology fitness

Ontology fitness is the quality of interaction to deploy learners' mental processing toward the core of what is intended to be learned. Therefore, ontology-fit interaction, embedded within a task, should be as such that it excludes the irrelevant and superfluous feature. For this reason, it promotes more target-focused learning (Sedig, Liang 2007).

Learner fitness

Learner fitness is the quality of interaction to address the cognitive needs of a group of learners (Sedig, Liang 2007). This quality takes into account the fact that the cognitive potential of each individual is different.

Context fitness

Context fitness is the quality of interaction to address the requirement of the context in which the interaction is represented (Sedig, Liang 2007). For instance, VRs can be designed in the context of a game, which introduces its own requirement.

From the design point of view, these five fitness criteria can be used to evaluate the efficiency and efficacy of interactive software.

2.5 Multimedia

Multimedia, in general, refers to the category of media that utilizes different types of content such as text, audio, video, and game. Using different types of communication channels suggest a richer and more interactive environment. However, it also brings more complexity and less coherence.

Educators and practitioners have adopted multimedia as a mean to promote learner's involvement and encourage effective ways of learning. However, many researchers question the potential of multimedia in making a difference in learning outcomes (Clark 1983; Martin, Rainey 1993). Clark and Mayer (2003) assert on the influence of the design of contents. They claim that the delivery method of instructional content is not that important, and what really matters is the "designing of the lesson materials with the right instructional methods".

Many researchers describe the application of a multimedia system as a way of facilitating different delivery methods such as text, audio, and video, and they assert their positive influence on the learning and knowledge transfer (Jacobson, Spiro 1995; Kozma 1991). They primarily believe that multimedia can support learning by providing logical organization of content and multiple methods of presenting it which results in efficient cognitive mapping (Jonassen 1990; Tulving 1983).

Moreover, other researchers emphasize the adaptive nature of multimedia as a way of delivering more customizable contents (Duffy, Knuth 1990) that may address different learning styles of individuals (Keller 1983).

A particularly applicable study that proposes a design framework is the cognitive theory on multimedia learning by Mayer (2001). Mayer (2001) suggests that learners use separate channels to process visual and auditory information and each of these channels has a limited processing capacity. He, also, has identified three multimedia-related cognitive activities, selecting, organizing, and integrating, which facilitate the process of learning.

To summarize, multimedia can potentially provide different representations and methods of delivering content. As a result, it provides opportunities for addressing more audiences. However, design of such a media should be systematically supplied to facilitate thoughtful reasoning and improve learning.

2.6 Knowledge

Knowledge is a complex and abstract phenomenon. Understanding the nature and the scope of knowledge as well as determining means of its efficient communication are the primary objectives of any learning environment. To design a software environment which facilitates knowledge construction, the first step is to understand what the knowledge is.

Knowledge is a blend of information, context, interpretation, and experience (Davenport, Prusak 1998). It is not a static or stand-by phenomenon, but an ongoing process by which knowing takes place (Sveiby 1999). In this sense, action appears to be the key element of knowledge (Nonaka 1994).In a similar definition, Nurmi (1999, p128) points out the dynamic nature of knowledge as something that is "acted upon", in contrast with the passive nature of information.

Knowledge is the top of a hierarchy which through its creation, emerges from data and information (Ackoff 1989, Zeleny 1987). Within this hierarchy, data is the basic element or raw material with high entropy, no organization, and no meaning. Information, however, is more contextualized and more meaningful. The meaning of information is the result of grouping and organizing data as raw material, which creates a form, structure, or pattern. On the other hand, information can be extracted as patterns or classified into groups. Transforming from data to information and vice versa can be facilitated by computers to provide visual and numerical understanding of phenomena.

In brief, data and information are both passive. They are both entities outside the human mind. One is just more complex than the other. Knowledge, conversely, is active with the trace of human mind in its formation (Prusak 1999). The misconception among designers of learning software that knowledge is simply the same as information but in more complex form, has resulted in poorly delivered educational environments. These environments are mostly restricted to delivering information, in contrast to transferring knowledge.

Although knowledge may have a different nature than information, it still evolves from information. Thus, information is the main ingredient of knowledge. One can say that the mind constructs knowledge out of information through a series of cognitive activities. Complying with this line of thought, the appropriateness of the design for the computer-aided learning environments can be viewed in terms of its potentials toward support such activities.

Through the process of transforming from information to knowledge, some information simply links together and creates a simple form of knowledge. Nonaka (1999) refers to this simple form as “explicit knowledge”. In his definition, explicit knowledge is a type of knowledge that can be expressed in terms of information or data. In this case, few cognitive processes are applied on information to form explicit knowledge. These cognitive processes can be as simple as capturing or memorizing some of the information without further analysis or interpretation. Therefore the nature of information is mostly intact and consequently can be restated in terms of information or data.

On the other hand, information can be changed in nature to form more personalized, contextualized, complex, and holistic structure. Nonaka (1999) refers to this type of knowledge as tacit knowledge. He further asserts that tacit knowledge is hard to be forced into formats and thus, it is difficult to communicate (Nonaka, Konno 1999). Nonaka (1994) emphasizes the presence of cognitive elements in the shaping of tacit knowledge. For that reason, tacit knowledge represents the knowledge of individuals. It is what individuals conceptually get out of external information. Therefore, it is an individual insight or personal interpretation of phenomena.

Sabherwal and Becerra (2001) assert that communication of tacit knowledge is possible through sharing among individuals and the conversion of knowledge from tacit to explicit (Lam 2000). From a design perspective, a learning environment can provide tools to support users' communication and knowledge transfer.

From the literature review of knowledge, it becomes clear that an individualistic approach to learning appears to fall short in the development of the type of knowledge which can be only constructed through sharing knowledge among individuals. From the design points of view, a learning environment becomes more efficient if it supports individual-individual interaction. The next section explores how social interaction aids learners in constructing knowledge.

2.7 Collaborative learning

Collaboration is a process through which participants work together toward constructing shared knowledge (Andriessen, Baker & Suthers 2003; Roschelle, Teasley 1995). The emerged shared concepts, which are outside the individuals' vision, are constructed through identifying the distinctions and exploring the differences among participants' ideas (Gray 1989).

There has been a great amount of research asserting the positive role of collaborative learning (Lally, Laat 2002; Mulder, Swaak & Kessels 2002; Teasley, Roschelle 1993). Collaborative learning portrays a new picture of learning environments as a knowledge construction space in contrast to information delivery space. Therefore the primary goal is not to simply transfer a specific knowledge to learners' head, but to support them to construct their own knowledge and share those among each others. In this sense, not only individuals build their own knowledge but also are exposed to others' views and perspectives. Collaborative learning introduces a series of concepts which are briefly explained in the following sections.

2.7.1 Shared mental model

In addition to the concept of individual mental models, shared mental models were first introduced by Cannon-Bowers and Salas (1993). It is mostly concerned with knowledge as a distributed construction within a team. Team mental models encompass the following concepts (Mohammed, Dumville 2001).

Information sharing

Information sharing enhances the decision making process by sharing pieces of information. The information sharing relies on the basic principle that a group produces better decisions and infers more tangible judgments than individuals.

Transitive memory

Transitive memory views memory as a social phenomenon. In this view, individuals exploit each other's memories as an external resource. Therefore, members of a society or group extend their cognitive capacity by relying on each other's memory.

Group Learning

Group learning is characterized as a perpetual process of learning within a group. It revolves around asking questions, seeking answers, trial and error, and discussing errors. It also is an outcome that manifests itself in the performance of a group.

Summary

As previously mentioned, mental models represent our view of the world. We perceive things and make sense of them either through observation, interaction, or the influence of other people's explanation and experience. Further, our power to formalize things internally, in addition to our perceptions, helps us toward constructing our mental models. Therefore, mental models are the end product of how we individually understand the world. This makes it particularly important in terms of reasoning and the interpretation of ideas and phenomena, as we refer to our individual knowledge (mental model) to perform those tasks.

Moreover, our mental model can be influenced by knowledge of societies. This brings up the concept of knowledge sharing and collaborative learning. Individuals can enhance their mental model when they face a conflict with other members of society, are exposed to their ideas, explore differences, and find new solutions. Following this idea one can conclude that shared consensuses are usually stronger and supported by a stronger body of knowledge (knowledge of group).

However, creating such a society to facilitate learning and enhance the overall performance of group should be carried out systematically. Davenport (2001) (Davenport, Grover 2001) brings up the notion of "communities of practice" which refers to a society of people whose interests and goals link them together and they share their knowledge. Durrance (1998) proposes four characteristics for the society, which promotes knowledge sharing. First is a watching mechanism, second is creating an environment of trust, third learning by doing, and last allowing reflection and interpersonal exchanges.

In terms of design, the ideas explored in this chapter provide a theoretical foundation promoting a systematic approach toward implementing online learning environment.

Chapter 3
High-level design of online learning environments

Complying with the theoretical principles (see Chapter 2) and along with the primary objective of providing an integrated online learning space, this chapter explains the design of two online environments implemented for this research. It also explains how the principles of visualization and interaction are utilized to promote explorative and constructive modes of learning mathematics.

Moreover, this chapter presents the details of the user interface as well as the visual components within the learning environment.

3.1 Mathematical topics

The 13 Mathematical topics chosen for this research are considered to be the key topics of mathematics in grades 10-12 and early levels of university mathematics. They are observed by teachers as fundamental topics in mathematics requiring a wide range of mathematical proficiencies to learn. Furthermore, these topics are chosen in conformity with the Ontario Mathematics Curriculum expectations .

The explored topics include trigonometry, exponential and logarithmic functions, graph sketching, inequalities, rate of changes, absolute values, polynomials, probability, quadratic functions, rational functions, sequences and series, and systems of equations.

For each topic a series of interactive visualizations is designed and developed. The designed interactive visualizations are embedded in two integrated online environments. The next section describes the conceptual model developed as the basis for designing the two environments.

3.2 Conceptual model

As described in chapter 2, the theoretical principles of the learning can be utilized to form a foundation for the development of a more efficient learning environment. This section incorporates those principles within a conceptual model. The conceptual model is presented in the Figure 3-1.

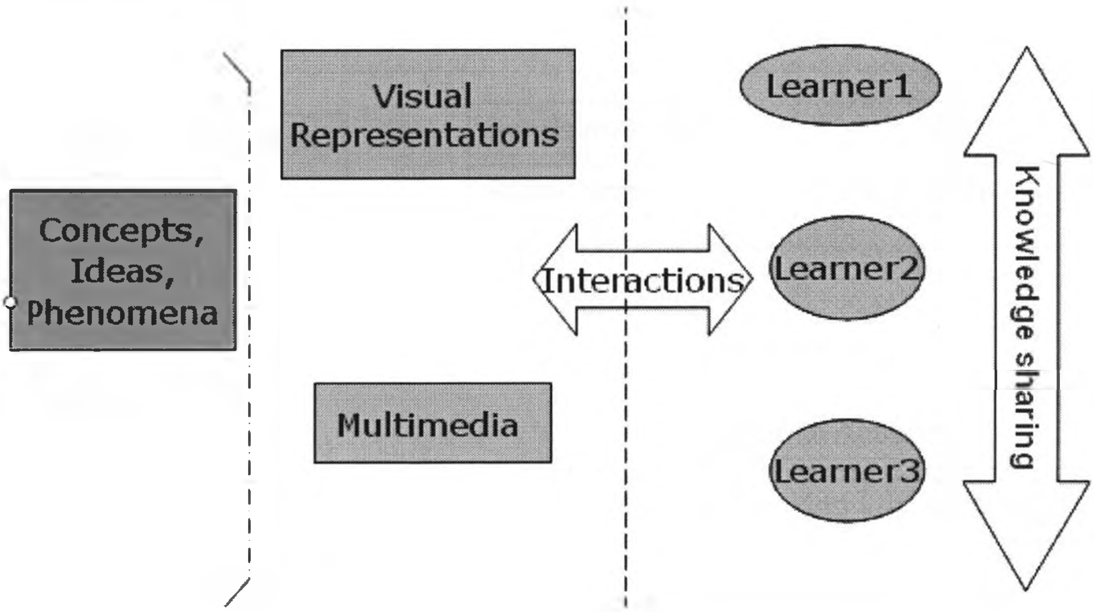

Figure 3-1 the conceptual model proposed for the design of leaning environments

This model consists of five components listed as learners, visual representations, interactions, multimedia, and knowledge sharing. The first and the most central component is the learner. As such, learners are considered to be the major contributors to enriching the content of the environment. That is, learners interact with visual representations of the environment and build their own views of the presented concepts and ideas.

Then, they share their own understanding with each other using the tools provided within the environment. The environment captures the shared ideas and saves them into a data repository. Therefore, the environment dynamically develops a rich content comprising different points of view and the diverse perspectives of its users.

The design of visual representations should be in accordance with the principles of the learning theories to effectively lead learners toward acquiring the knowledge (see Chapter 2). In this sense, the model expands to embody those principles. Figure 3-2 describes the extension of the conceptual model.

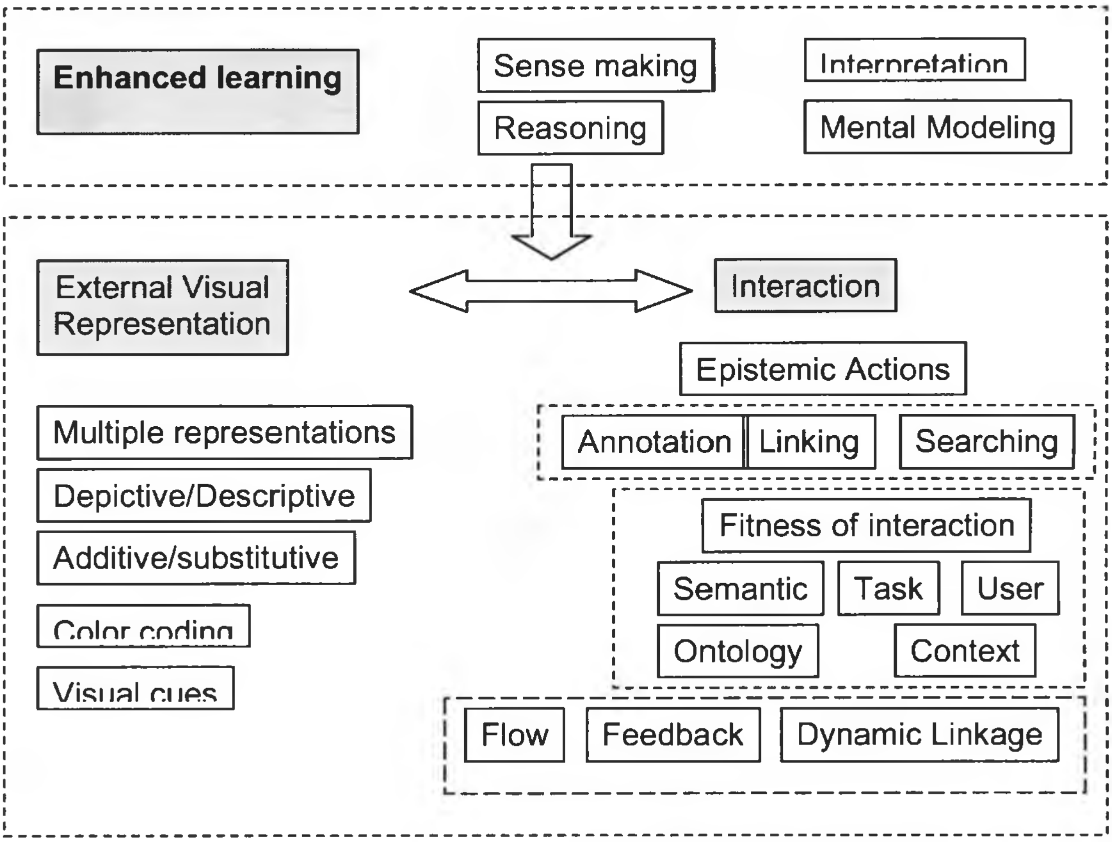

Figure 3-2 the extension of the conceptual model with the focus on individual learning

This model integrates the principles of visualization and interaction (see Chapter 2). At the top of the model, enhanced learning is considered as the most primary objective. In addition; sense making, reasoning, interpretation, and mental modeling are categorized as the components of learning and as decisive drivers influencing the design of interactive visual components. The model takes into the account the importance of providing multiple visual representations, fitness of interactions, as well as supporting learners toward performing epistemic actions. The outcome of this extension of the model is a series of well designed interactive visualizations which comply with the principles of human learning. However, this extension merely supports the individualistic style of learning. That is, learners are provided with interactive visualizations to make their own understanding of the presented concepts and topics.

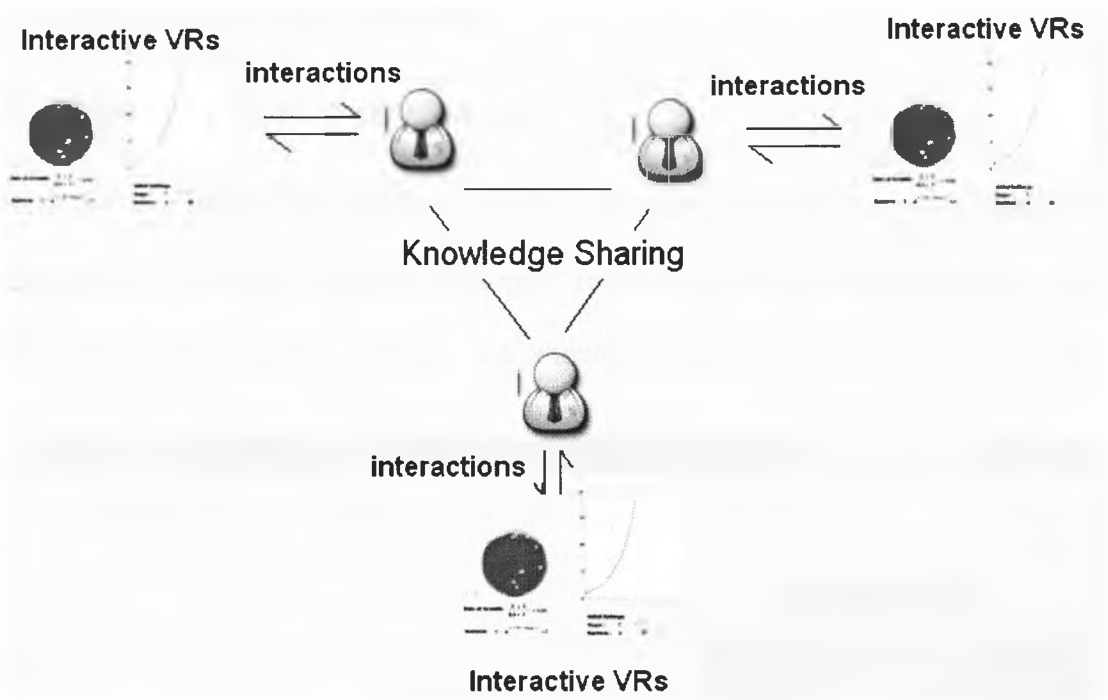

Figure 3-3 the extension of the conceptual model with the focus on collaboration and knowledge sharing

To further augment the effectiveness of the learning environment, the model considers provisioning of tools by which users can share their understanding and interact with other users of the environment. This view of the model is depicted in Figure 3-3.

The outcome of this view is an integrated environment which allows users to construct their knowledge and share it among each other. In this sense, this view of the model promotes a more collaborative learning.

Moreover, multimedia and instructions embedded in the model are considered to enrich the environment (See Chapter 2).

3.3 Description of the online environment

Inspired by the conceptual model, two learning environments are developed and made publicly available. These two environments are as follows:

- Digital Windows into Mathematics
- Mathematics Preparedness Program

Both learning environments provide interactive visual representations to facilitate learning of the chosen mathematical topics. In addition they provide multimedia and tools for knowledge sharing. Figure 3-4 shows elements of these two environments.

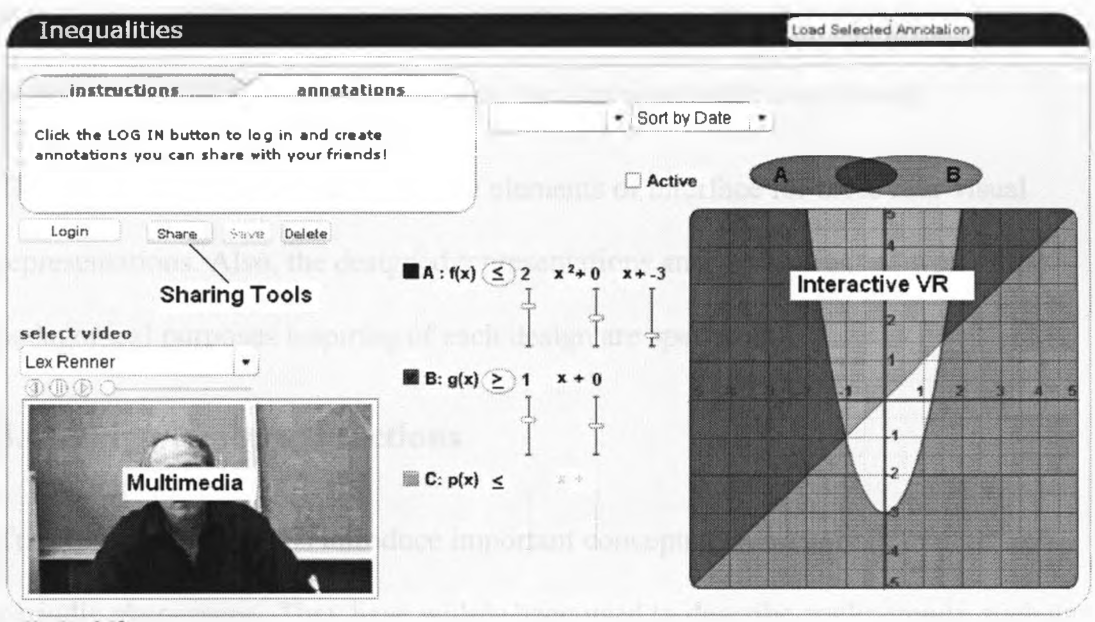

Figure 3-4 elements of the implemented environment

In each environment, multiple visual representations illustrating a mathematical topic are designed and implemented to support enhanced reasoning and interpretation (see Chapter 2). Also, pedagogical goals for the design of each interactive VR are taken into account and appropriate interactions and visual representations (see Chapter 2) are designed and implemented.

Among thirteen topics, 4 topics are chosen and their details are presented in the current thesis. These topics are as follows:

- Trigonometry
- Exponential and logarithmic functions
- Graph sketching
- Inequalities

While the four chosen topics are rich in pedagogy, the provided visual representations for them are sufficient to demonstrate the key design strategies and rationales. Also, they incorporate the relevant elements of the conceptual model clarifying how the pedagogical objectives are mapped into the actual software components.

The rest of this chapter elaborates the elements of interface for these four visual representations. Also, the designed representations and interactions as well as the pedagogical purposes inspiring of each design are specified.

3.4 Trigonometric functions

Trigonometric functions introduce important concepts related to both triangle and periodic phenomena. They have widely been used to describe cyclic trends, such as physical phenomena (e.g. alternating electrical currents), environmental phenomena (e.g. the total radiant energy received by the earth from the sun in periods of time (TSI)), or social phenomena (e.g. cyclical nature of recession and economy).

This research explores the three major trigonometric functions of the sine, the cosine, and the tangent. In that regard, the following most common forms of three trigonometric functions are presented.

- $Y = aSin(bx - c\pi) + d$
- $Y = aCos(bx - c\pi) + d$
- $Y = atan(bx - c\pi) + d$

Each trigonometric function includes four coefficients. The state and the effect of each coefficient can be explored through the depictive representations of trigonometric graphs.

Moreover, each coefficient introduces a mathematical concept, which can also be understood from the states of the graphs. These mathematical concepts are listed as follows:

- *Amplitude*
- *Phase shift*
- *Altitude*
- *period*

The *Amplitude* is in direct relation to the coefficient “a” and represents how stretched the graph of trigonometric functions can get vertically.

The *Altitude* is in direct relation to the coefficient “d” and represents how far the graph of trigonometric functions can move vertically.

The *Phase shift* is in direct relation to the value of coefficient “c”. It represents how far the graph of trigonometric functions can be shifted in either direction of left or right.

The *period* is defined as the length of the time through which a trigonometric function repeats itself. For example, in Figure 3-5 , within each “π” time the sine function repeats itself once.

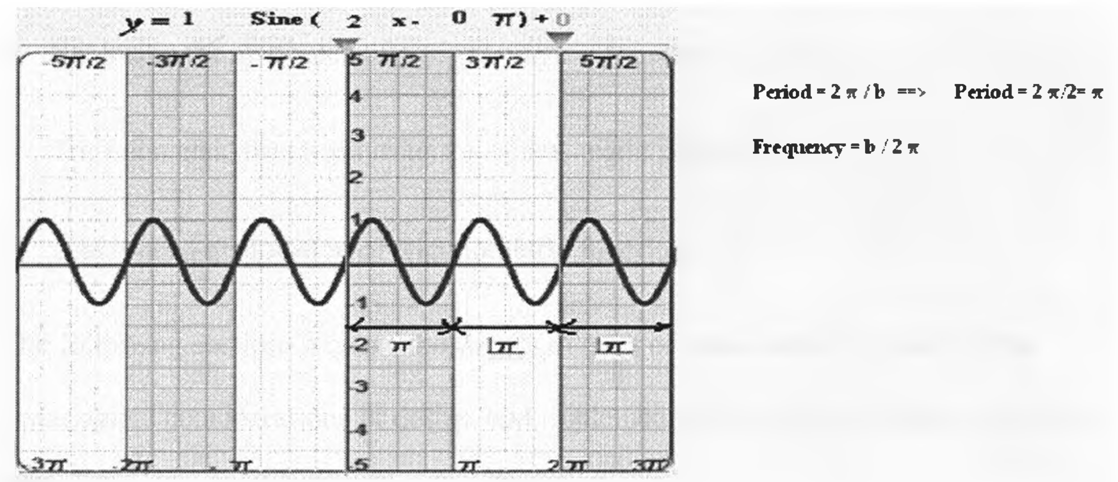

Figure 3-5 Trigonometric function and the concepts of period and frequency

On the other hand the number of repletion in each “2 π” time is defined as frequency. Additionally, the period and frequency are related to the value of coefficient “b” as follows:

- $Period = 2\pi/b$ *inverse relation*
- $Frequency = b/2\pi$ *direct relation*

The current research explores the above concepts through implementing a set of representations and interactions. Abstract concepts such as amplitude, altitude, period, and phase shift are investigated through the provision of multiple visual representations and interactions which amplify deeper and more effective learning (see Chapter 2). Provided visual representations and interactions are intended to communicate different pedagogical aspects of trigonometric functions. Those aspects are listed as follows:

- Drawing the curve of a trigonometric function
- Exploring the relationship between the coefficients and the graph of a trigonometric function
- Exploring the attributes of a trigonometric function
- Trigonometric functions in terms of real world phenomena
- Task-based exploration of trigonometric functions

The following sections explain the details of each of these aspects as well as the pedagogical considerations in design and implementation of the provided interactive VRs.

3.4.1 Drawing the curve of trigonometric functions

The pedagogical goal of this representation is to aid learners to understand the relation between a trigonometric function and its curve. That is, they are expected to contemplate how a trigonometric function can produce a curve. In this sense, they are required to perceive a trigonometric function as a rule which determines how its curve would be drawn. Also, they are encouraged to recognize that a curve is constructed out of a series of connected points, which their relation is defined by the function.

An application called "Programmable Points" is designed to explore the above ideas. This application allows users to draw a trigonometric curve by understanding the role of its function and realizing how the curve is constructed out of connected points.

Interface of the programmable points

In order to support exploring the above ideas the interface as presented in Figure 3-6 is designed and implemented.

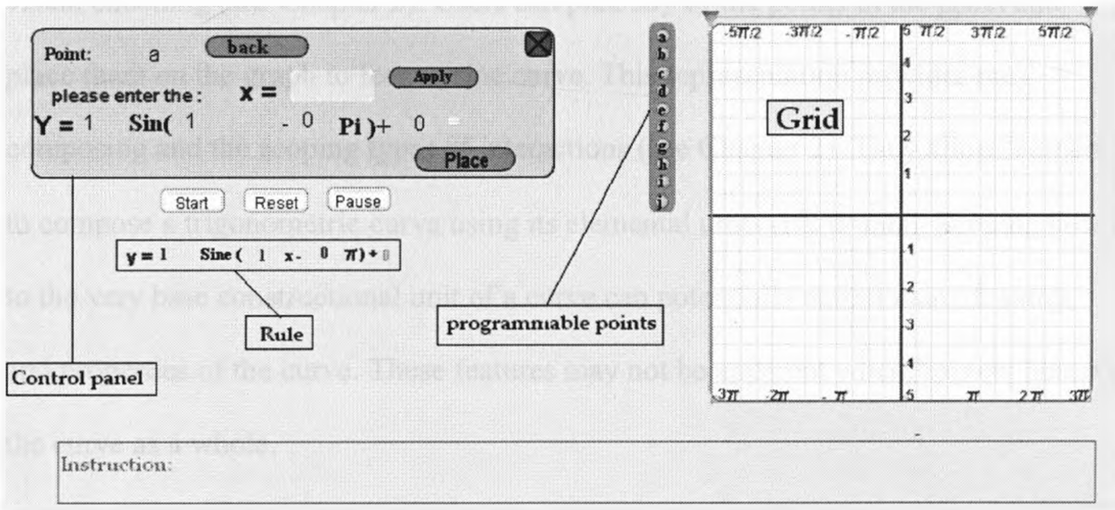

Figure 3-6 interface of the programmable points

The interface provides means by which users can actively participate in interacting with points as the elemental units of a curve and maintain a bottom-up understanding of how a trigonometric curve is constructed. The components of the user interface are the grid, the programmable points, the control panel, and the rules. These components are detailed in the following sections.

Grid

The grid is designed as a rectangular shape, which includes vertical and horizontal axes. The horizontal axis is scaled based on gradient values and the vertical axis is scaled based on integer values. The grid holds the position of points after they are programmed.

Programmable points

A total of ten points are designed and made interactive to be placed on the grid. The points are listed alphabetically and placed in a panel. The alphabetical arrangement of points makes it easy for users to keep track of them on the grid and provides a simple visual encoding (see Chapter 2). Users can pick any of the points in the panel and place them on the graph to form a sine curve. This representation provides the composing and the scoping types of interactions (see Chapter 2). They allow learners to compose a trigonometric curve using its elemental units (i.e. points). Scoping down to the very base constructional unit of a curve can potentially reveal many features and properties of the curve. These features may not be apparent when learners observe the curve as a whole.

Control panel

The control panel is designed as a utility unit, which allows users to calculate the position of a selected point on the grid. To accomplish this task, the interaction is designed in the form of a conversation mechanism. That is, it provides a text box allowing users to insert a value for the input variable of the function. Also, it maintains immediate and quantitative feedback of how the trigonometric function is applied to the selected point by providing the output value of the function.

The panel is composed of three buttons (i.e. start, pause, and reset). These buttons are served to facilitate an animation type of interaction (See Chapter 2). They allow users to generate an animated curve, presenting the gradual formation of the curve. The animated curve cut across the placed points and provides visual feedbacks. Also, the animated feedback highlights the relationship between points by connecting them together. Moreover, it provides a sense of direction by starting from and finishing at the specified points. In this sense, the animation supports users through performing some epistemic actions such as searching and linking (See Chapter 2). Also, the users' flexibility in pausing and resetting the graph allows them to explore and control the pace of their investigation.

Rule

The rule is considered to be the chosen trigonometric function, which provides constraints on how and where each selected point should be positioned on the grid. The constraint is intended to emphasize the role of the trigonometric function in placing points on the curve. The interface maintains substitutive representations to communicate the intended underlying logic (See Chapter 2)

How to program points

In order to program a point, users first select one of the points listed in the panel using the mouse pointer. Clicking on a point moves the selected one to the control panel as described by Figure 3-7

In the next step, users can insert an input value in the text box and press the *"apply"* button to apply it. The value will be shown in the sine function and an output will be calculated. Also, prior to placing the selected point on the curve, users can change and edit the input value of the point and calculate different outputs. As an output is generated immediately, it allows users to adjust their decision by making sense of the produced values.

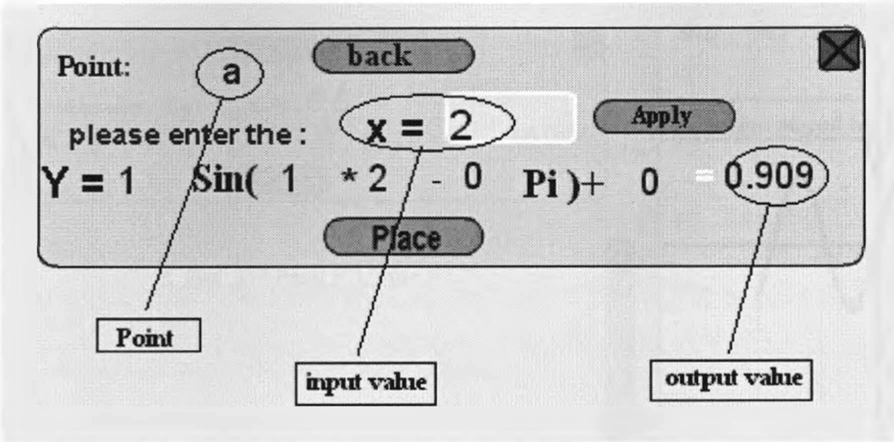

Figure 3-7 control panel used to assign different values for the point

The last step is to place the selected point on the grid, which will move the point based on the calculated position. Clicking on the *"place"* button, which places the point on the grid, performs this task.

By delaying the time between the calculations of the point's vertical and horizontal coordination and the task of placing it on the grid, users are intended to thoughtfully engage with the process of constructing the curve. First, they make sense of the coordination's values of the points, and then they learn how the function is applied to calculate those values. Second, they visually observe how the trace of points defines the formation of the curve.

At any time, the user can run the animation by clicking on the "start" button to see how the curve will be formed by passing through positioned points. This feature provides a visual hint by picturing the end result of the calculation and supports learners in performing epistemic actions of searching and linking.

Also, users can reset and stop the animation at any time. The Figure 3-8 shows how eights of these points are programmed and placed on the curve, and how the animated curve is run through them and stopped at point "d".

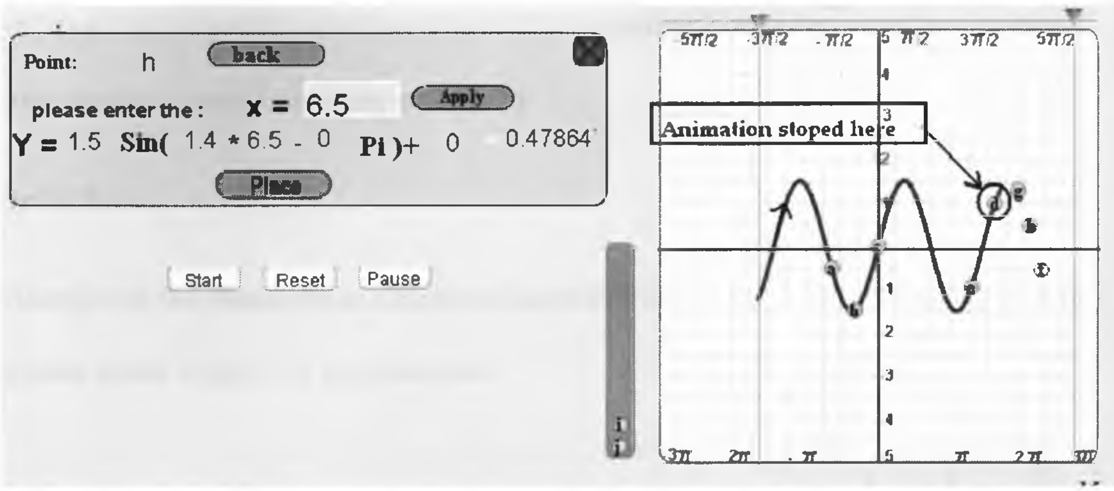

Figure 3-8 running example of the programmable points

Instruction

This portion is composed of a series of instructions, which directs users toward exploring different aspects of the curve. Also, it is considered to be filled by experts to extend the pedagogical aspects of the application and provide direction for learners to efficiently explore the concepts. For example, an instruction for this software may suggest trying the points with the same absolute value such as 2, -2, which provides an insight to the symmetric property of sine curves (i.e. Sine (-x) =- Sine (x)).

Further, purposeful direction will allow learners to gain knowledge of some properties of a sine curve such as the Maximum points, the minimum points, the altitude, and the amplitude.

3.4.2 Exploring the relationship between the coefficients and the graph

In this mode, the pedagogical focuses are on investigating the relationship between the value of a specific coefficient and the graph of the trigonometric function. In this sense, learners are required to understand how changing the value of each coefficient of a trigonometric function will affect the state of the curve of that function. These relations are intended to be investigated by providing a series of visual components and allowing users to interact with them.

Interface

Along with the pedagogical objectives described in the previous section, the interface shown in the Figure 3-9 is developed.

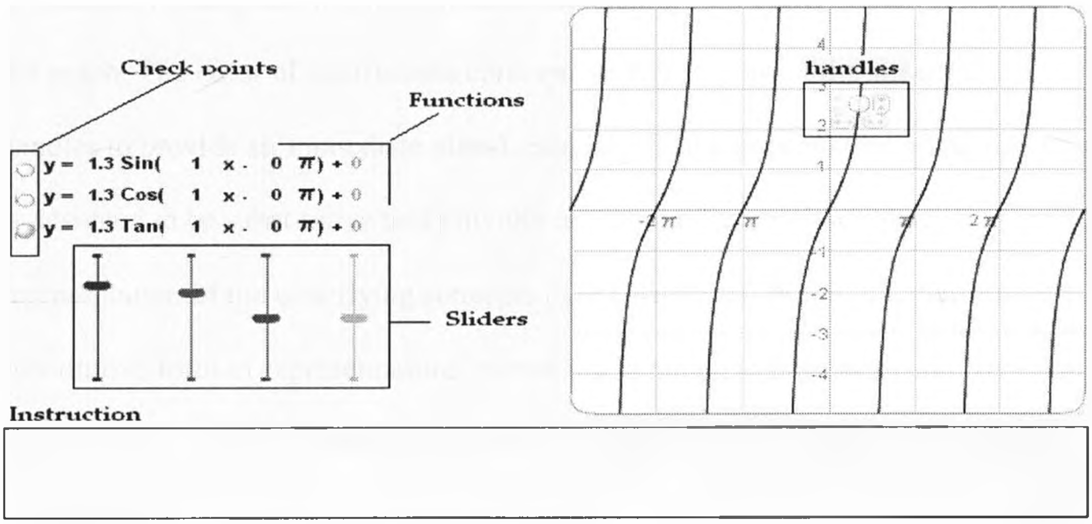

Figure 3-9 Trigonometric functions

The design of the interface is geared toward assisting users to make a clear comparison between the outcomes of choosing different values for coefficients. The components of the user interface are the check points, the functions, the sliders, and the handles. These components and their functionalities are explained in details in the following sections.

Check points

The check points allow users to select one of the trigonometric functions and observe its graph. They also allow users to dynamically observe and make comparison between the graphs of different trigonometric functions. In this sense, users can pick any of the functions, then change the values of its coefficients and compare them with another trigonometric function. To provide users with an immediate visual encoding the selected checkpoint is made highlighted.

Functions and curves

The listed functions are composed of three trigonometric functions of the sine, the cosine, and the tangent. Each function includes four coefficients, whose values dynamically change as the result of users interacting with either sliders or handles on the graph. The color of coefficients corresponds to the color of the sliders and the handles to provide an immediate visual encoding. The representation of the function is designed to be substitutive and provides a quantitative and descriptive representation of the underlying concepts (See Chapter 2). While functions provide a substitutive form of representations, curves are intended to present an additive form of representation allowing learners to clearly compare among different outcomes (See Chapter 2).

Sliders

Four sliders are implemented and designed as interactive mediums. They allow users to dynamically change the coefficient values of the functions by substituting new values. In addition, it allows users to explore depictive representations of functions in the form of their graphs. Sliders are geared toward providing a concept manipulation interaction (See Chapter 2) as they allow users to alter the values of functions and observe the outcome in the form of visual feedbacks. Providing both descriptive and depictive feedbacks promotes a flexible way of mathematical reasoning (see Chapter 2). Moreover, sliders provide an on demand interaction mechanism by which users can explore the visual representations in their own preferred pace by controlling when and where to stop the process.

Handles

The handles, similar to the sliders, are intended to operate as interactive mediums. Each handle allows users to investigate the effect of a coefficient on the graph of a trigonometric function. This interactive medium, however, is designed as an object manipulation interaction by which users are able to interact with the actual graph as a visual object (See Chapter 2). In this sense, they differ from sliders that primarily interact with the values of coefficients. Using handles, learners are able to manipulate the state of the curves. However, the design of this interaction medium limits the users' control, in that it does not allow them to aimlessly drag the graph across the grid. Instead, the range of the motion defined for each handle corresponds to a specific coefficient and is limited by the boundary of the grid. This feature prevents users from being exposed to superfluous and irrelevant contents and presents a more ontologically fit interaction (see Chapter 2).

The flow and linkage of the interactions, in this case, are designed as a bidirectional and harmonious model by which the user can interact either from object to concept (i.e. using handles) or from concept to object (i.e. using sliders). These bidirectional interactions are intended to provide more meaningful and deeper understanding of the inherent concept.

Instructions

The instructions provide directions and allow learners to touch on areas which are not straightforward or require coaching and support from instructors. There are situations that users may neglect to explore, or consider as being trivial while there are, conversely, rich in pedagogy and significant to the level of their learning.

For instance, an instruction may provide direction toward comparing the sine curve with the cosine curve that has the same coefficient values. It can be done by asking the user to make the two curves equal by changing one of the coefficients. This instruction allows users to understand the inherent concept, which is described by the following formula:

- *Sine (x) = Cosine (x+π/2)*

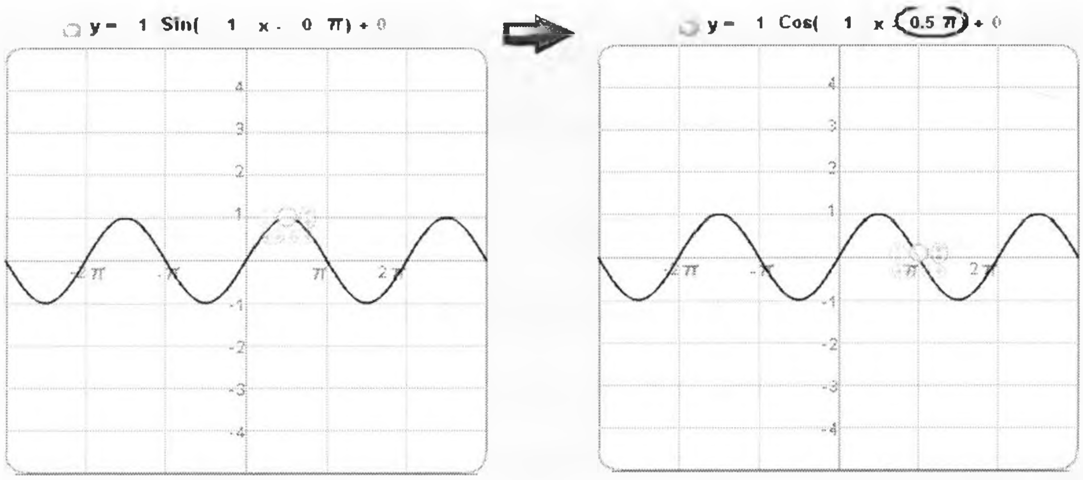

Figure 3-10 Relation between the sine and the cosine function

How to use the interface

To explore the relation between a coefficient value and the state of the curve, users can use either the slider or the handle which corresponds with that coefficient. Dragging the slider or the handle will change the state of the graph respectively. Users can immediately observe the outcome of their actions both substantively (changing the values of coefficients) and additively (changing the state of the curve).

3.4.3 Exploring the attributes of a trigonometric function

In exploring phenomena, learners are mostly asked to discover patterns and identify relations through interacting with visual representations. However, in the world of mathematics, learners are expected to take some further steps. They are expected to formulate those relations, generalize them, explore their reasons, and provide causal explanations of those phenomena. The generalization further can be applied to allow them to explain similar phenomena, and even devise new methods. Using previous versions of trigonometric functions, learners are provided with dynamic and harmonious representations, which facilitate identifying the relation between coefficients and the trigonometric curve. However, it does not provide either descriptive or depictive representations of why those relations existed or what the concepts are, which can be extracted to explain such relations.

For instance in the sine function of $Y = aSin(bx - c\pi) + d$, altering the value of "*b*", will cause the graph to expand or shrink horizontally. However, this behavior can afford deeper communicative level, if it provides more details about why it is happening. To do so, I have designed another version of the sine application, which allows users to be exposed to more concepts via both descriptive and depictive visual representations.

Interface

Figure 3-11 represents the provided interface and embedded visual component of this version. The focus of this VR is to promote deeper understanding of trigonometric functions in terms of patterns of changes as well as the inherent rationales that justify the state of the changes. Moreover, it provides more interactive visual components, which their effects on one another facilitate mathematical generalization and logical inference.

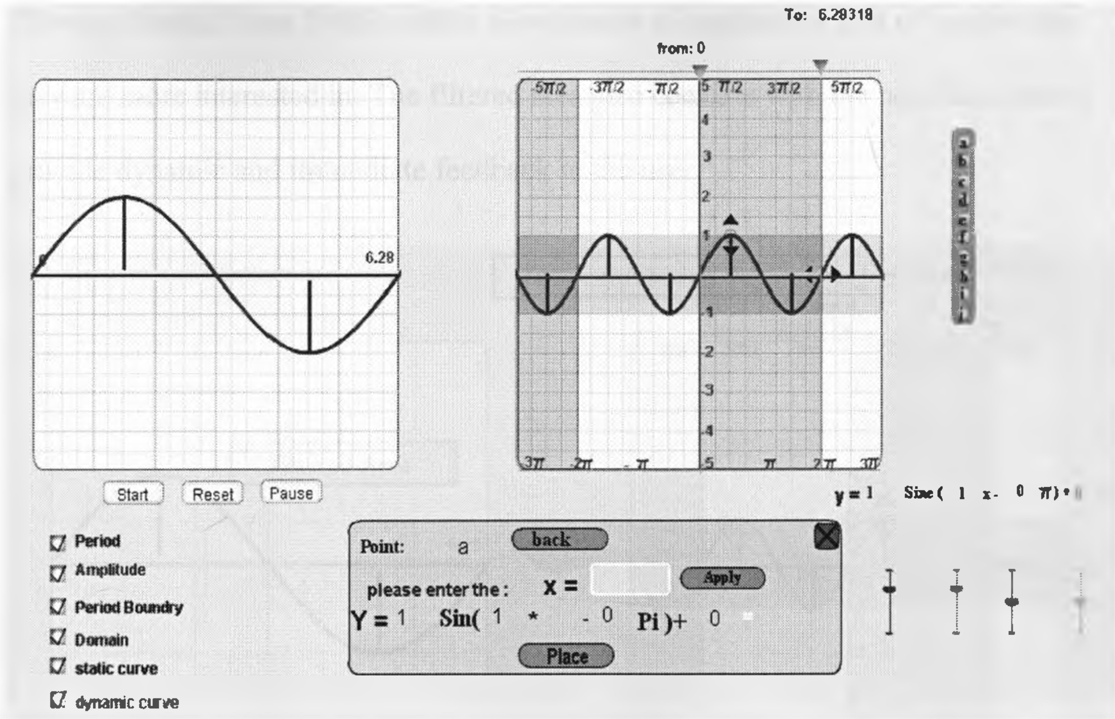

Figure 3-11 Interface for exploring the attributes of trigonometric functions

Furthermore, the interactive visualizations expose learners to various ways of approaching the underlying logic by implementing a wide range of interactions. The designed interaction types are filtering, probing, animation, composition, and annotations.

This version, while incorporating the features of the other versions, combines them and establishes a dynamic linkage among them. Hence, it offers a more flexible set of tools allowing users to have more control over investigating and exploring different aspects of trigonometric functions via performing different tasks.

The new interactive visual components and their functionality and features are described in detail in the next sections.

Filter the curve

In order to allow users to explore a sine curve and its attributes within a specific area of the grid, a filtering form of interaction is developed. It is designed as a range-based filtering (Sedig, Liang 2006), which allows users to explore the part of a curve that they are more interested in. The filtered area also coexists with the non-filter area to provide dynamic and immediate feedback to the user.

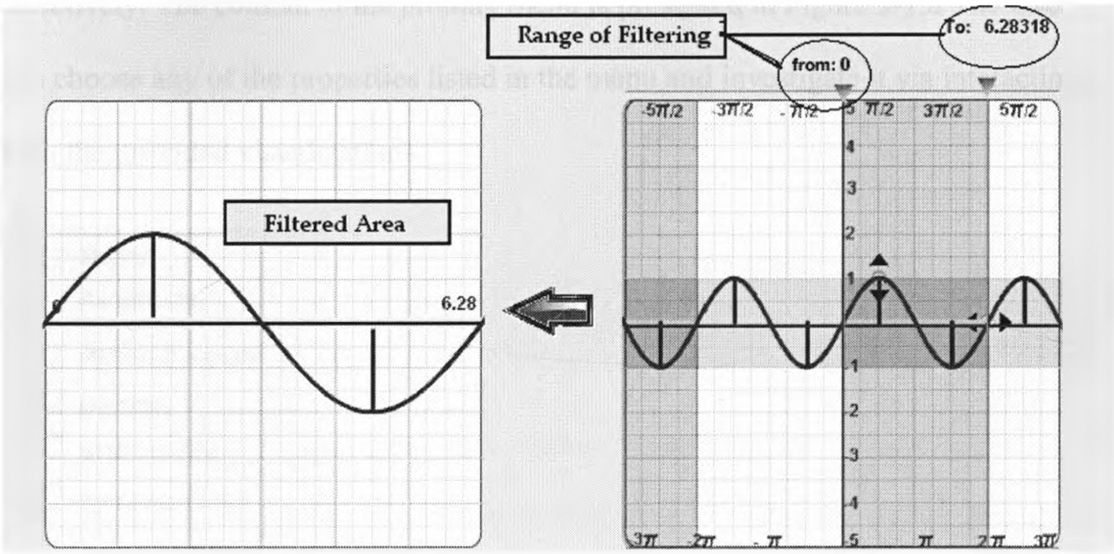

Figure 3-12 filtering as an interaction type

Having a range defined and filtered, users can explore curves, manipulate their coefficients, and observe the outcome in both filtered and non-filtered areas. Two sliders (i.e. "**from**" and" **To**") are implemented, and they define the range for the filtered area. The two sliders are illustrated in Figure 3-12.

To define the filtered area, users can drag any of the sliders and observe how the filtered area is captured in the second grid respectively.

Probing Menu

The Menu is designed to facilitate a probing type of interaction (see Chapter 2). It allows users to drill into chosen properties of a sine curve (Sedig, Sumner 2006; Sedig, Liang 2006). It first introduces a vocabulary of those properties and next allows users to add and remove them in the form of visual components.

Each item of the menu reveals a specific property of a sine curve. In this sense, the menu provides a "details-on-demand interaction mechanism" (Sedig, Sumner 2006, Sedig, Liang 2006) which help learners to explore properties of a sine curve selectively. The content of the probing menu is presented in Figure 3-13. The User can choose any of the properties listed in the menu and investigate it via interacting with the provided visualizations.

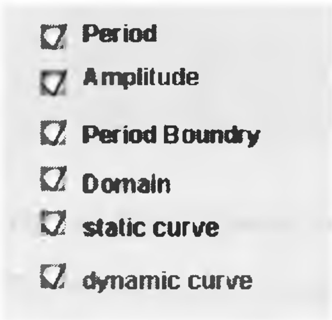

Figure 3-13 Probing menu

The period and amplitude visual representations

The period and amplitude are depicted visually as two straight lines. The line associated with the period is drawn horizontally, and the line associated with the amplitude is presented vertically. These two lines are depicted in Figure 3-14. The intention of this visual representation is to make the two concepts of the period and amplitude more explicit by making them visually available. In addition, two interactive handles are embedded at the end of the lines to allow users to interact with them.

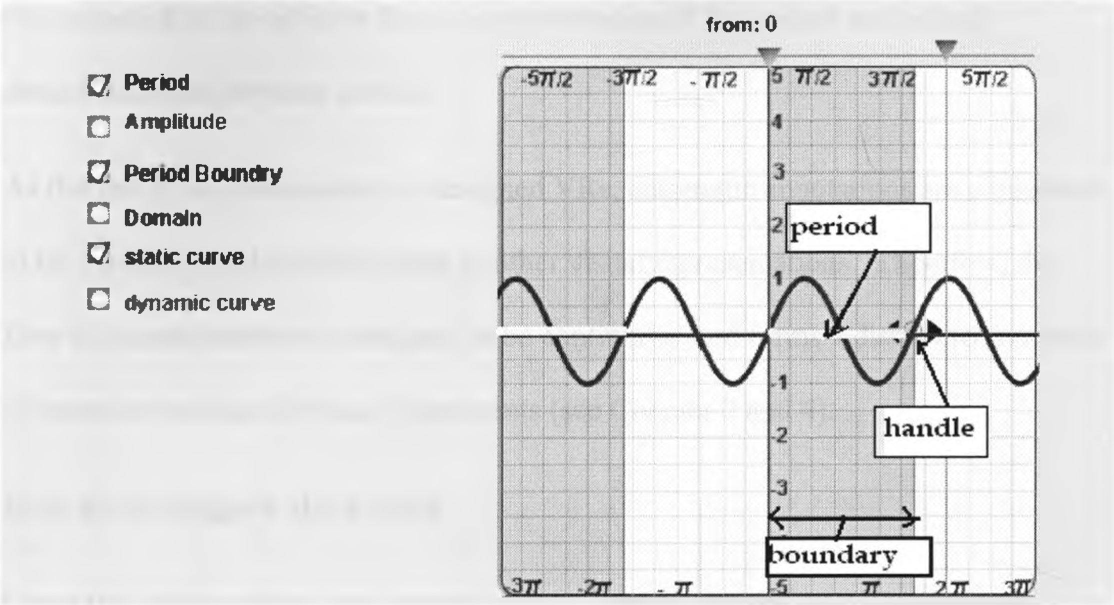

Figure 3-14 investigating the period of a sine curve

The two interactive lines provide additive forms of representation for the underlying concepts. They allow users to discern the effect of changing a coefficient value and to perceive it in relation to changing the size of the represented line. The logical association between the two concepts of the period and amplitude and the values of coefficients are implemented as a network of interactions. In this context, users are able to interact with either of a coefficient value or the line of the period and amplitude in order to investigate their effects on one another.

Annotation

The annotation is designed as an automatic and system provided interaction (Sedig, Sumner 2006). It explains the rationale of changing the two coefficients (i.e. "a" and "b" at $Y = aSine(bx)$ and their effects on the period and altitude respectively. It performs this task by exposing the user to the actual formula of the period and amplitude describing their relation to the two coefficients. The formula presents a substitutive form of representation. The substitutive representation is also accompanied by an additive form of representation of the period and altitude described in the previous section.

As the rest of the components in designed VRs, automatic annotations are considered to be dynamic and logically linked to other visual representations. Therefore, the flow of system reaction is designed to be continuous and forms a distributed network of reactions among all visual components (see Chapter 2 and 4).

How to investigate the period

Using the probing menu, the learner can focus on the features and visualizations intended to elaborate the period. Two options of "period" and "period boundary" are considered and are related to that property. Selecting or deselecting these options in the menu will show or hide the related representation in the form of a line or an area on the grid as described by Figure 3-14. At the end of the line an interactive handle is designed to provide an object manipulation form of interaction. The interactive handle allows the user to directly interact with the period rather than interacting with a coefficient. However, interacting with either of them, dynamically affects the other.

The user also can click on the handle to show or hide the automatic annotation describing the formula of the period. Figure 3-15 illustrates the use of annotation as a pop-up menu providing a quantitative feedback.

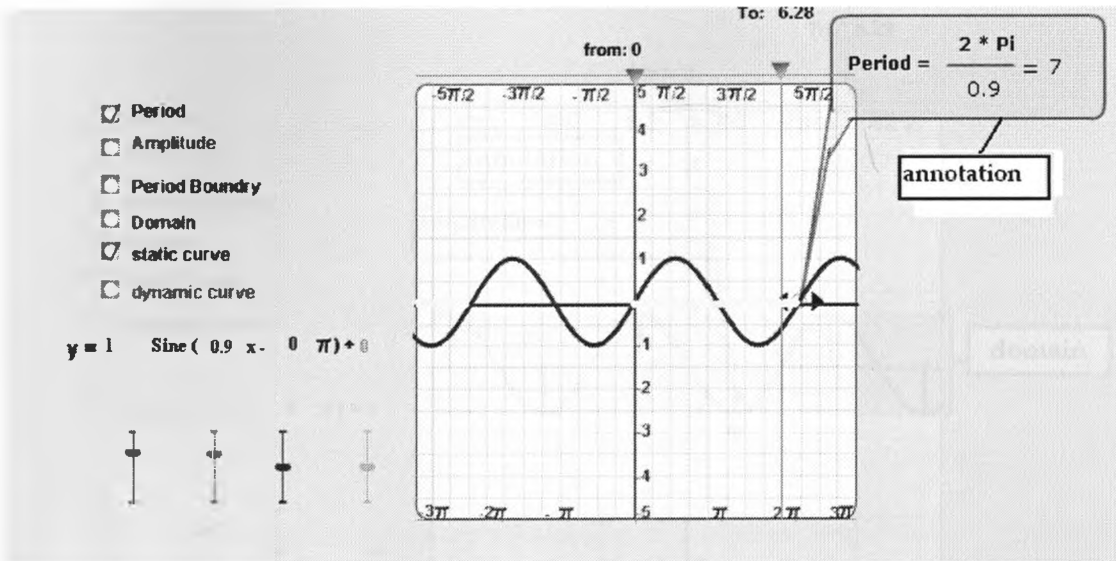

Figure 3-15 annotation showing the period formula

The values within this annotation change dynamically and correspond to the current state of the function and the actual size of the line of the period on the grid. It also provides a quantitative clue as to how the period is calculated mathematically.

How to investigate the Amplitude and Domain

Similar to the period, the amplitude is designed as a line with an interactive handle at its end. It facilitates altering the values of amplitude. Handles also provide an annotation describing the rationale and the mathematical formula. It represents the relation between the amplitude and the corresponding coefficient (coefficient a).

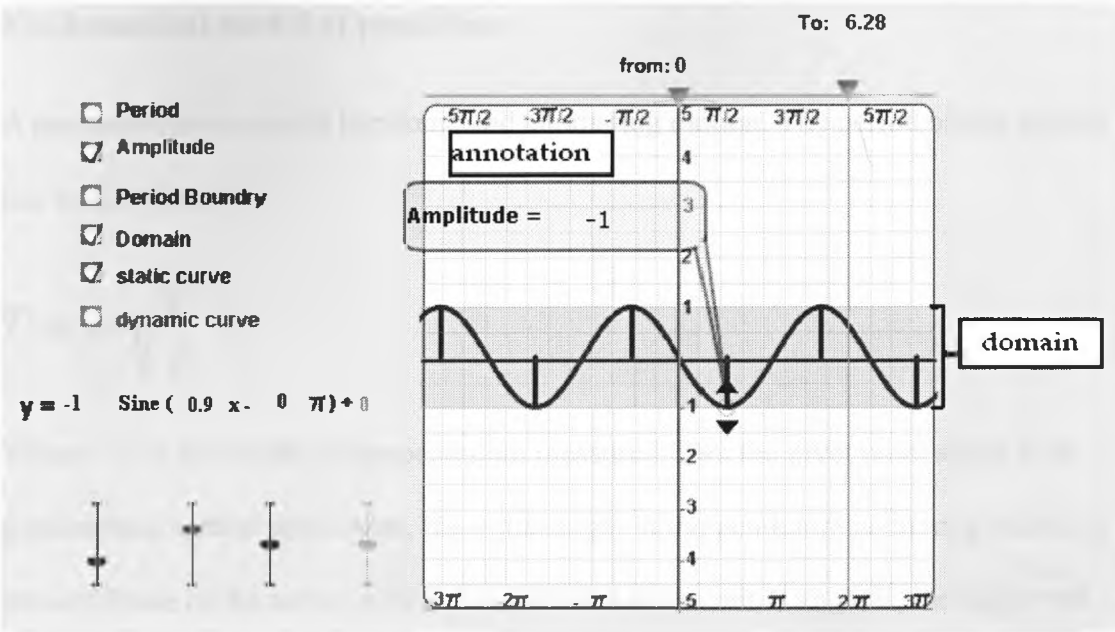

Figure 3-16 annotation showing the amplitude

Users can use the handle as well as the sliders to change the state of the graph. Provided interactions are illustrated in Figure 3-16.

3.4.4 Trigonometric functions and real world phenomena

To provide a sense of relevance in investigating trigonometric functions, this research presents a model of a pendulum as a physical phenomenon, the logic of which can be explained using either the sine or the cosine function. This model is served to provide a real life context to promote a situated way of studying trigonometric functions (Alessi, Stanley & Alessi 2001b; Gadanidis, Sedig & Liang 2004). Moreover, it is consistent with the line of thought which suggests that learners' deeper learning occurs, if they are exposed to a wider range of concepts presented across different context domains (See Chapter 2).

Mathematical model of pendulum

A pendulum has a simple harmonic and alternating motion. The period of this motion can be described as:

$$T \approx 2\pi\sqrt{\frac{\ell}{g}}$$

Where *"l"* is the length of the pendulum measured from the pivot point and g is the gravitational acceleration. Also, the initial angle of the pendulum is directly related to the amplitude of the curve, which means the wider the initial angle is, the bigger the amplitude gets.

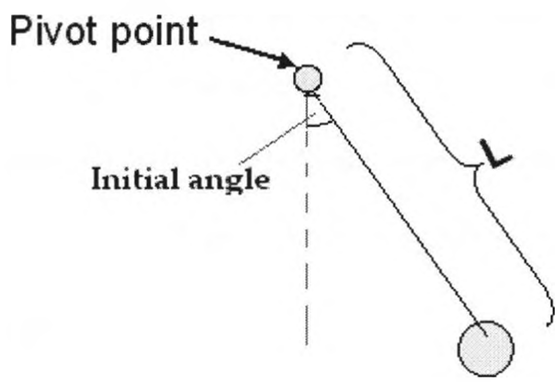

Figure 3-17 pendulum

Interface of pendulum

The representation depicted in the Figure 3-14 allows users to make sense of the cosine curve in respect to the pendulum as a physical model. It is also intended to provide a concrete example of how trigonometric functions are applied.

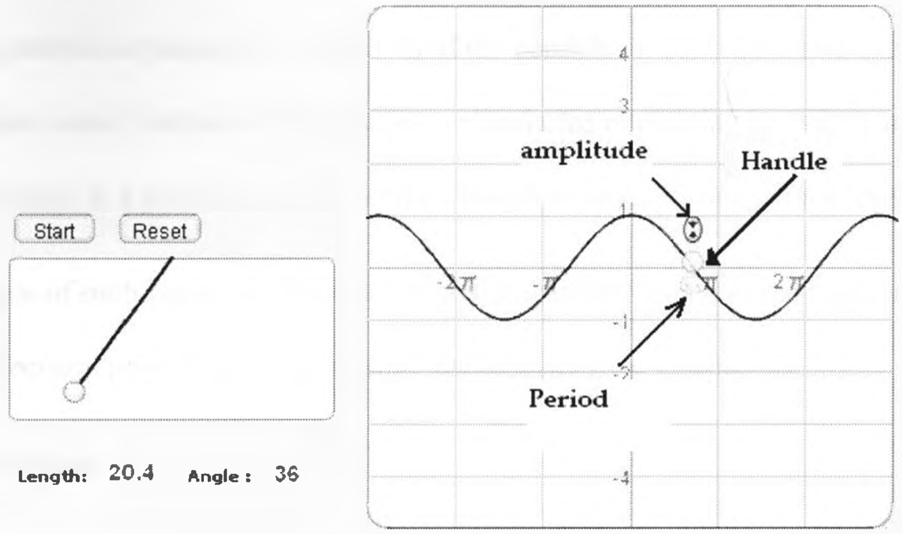

Figure 3-18 interface of pendulum simulator

The design of the interface provides a one-way interaction method by which users interact with the curve and observe the outcomes on the pendulum. The provided interactive medium is designed as a handle. It facilitates two ranges of motion corresponding to the pendulum's initial angle and the pendulum's length. The interactive handle provides a direct manipulation interaction (See Chapter 2).

Handles

Two handles are designed as interactive mediums. The upper handle is intended to change the amplitude of the curve while the lower handle manipulates its period. Since the period and amplitude of the curve are related to the length and initial angle of the pendulum respectively, altering them on the curve dynamically affects the length and initial angle of the pendulum. These orientations are designed in forms of both additive and substitutive representations. The substitutive representations are located below the pendulum area and quantitatively describe the length and the initial angle of the pendulum. The additive representations are depicted as the reaction of the pendulum to the adjustment of the state of the curve.

Users can use the handles located on the curve to initialize the length and angle of the pendulum prior to exploring the movement of the pendulum. The curve also serves as an immediate visual feedback that provides a visual clue explaining how the curve of motion will look if a pendulum with a certain length releases from a certain angle.

The presence of such feedback allows users to thoughtfully attend to the process of making a decision prior to performing their exploration (See Chapter 2).

Motion buttons

Two buttons are considered to start or stop the harmonic motion of the pendulum. The user can click on the start button and observe and explore the movement of the pendulum in the form of animation. It also generates a dynamic curve at the top of the initial and static curve. The dynamic curve describes the movement of the pendulum using a temporal representation. Placing both the static and the dynamic curve on the same grid is intended to support the comparative reasoning. These features are illustrated in the Figure 3-15.

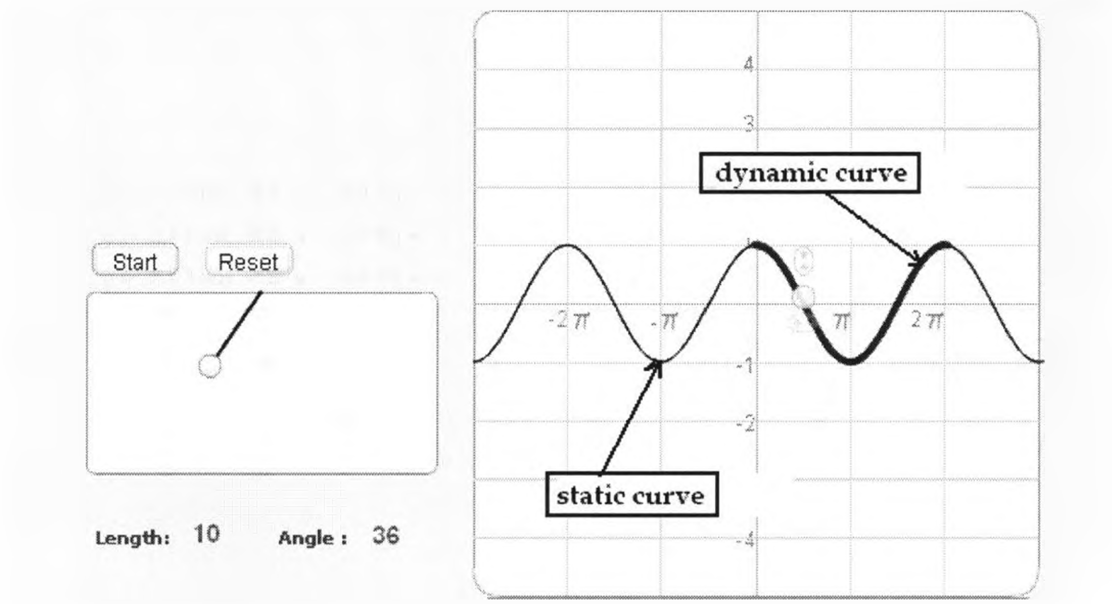

Figure 3-19 dynamic and static curve illustrating the motion of a pendulum

At any time users can reset the process, reinitialize the parameters, and explore the motion. This feature supports a flexible way of interaction that supports a trial and error approach of investigating a phenomenon (See Chapter 2).

3.4.5 Task-based Exploration

Learning by completing a task is another pedagogical approach of investigating a mathematical concept. Since there is a goal associated with a task it improves the users' attention and allow them to test their knowledge by completing a specified task. Performing tasks are also being used as the criteria to measure the quality of learning (Wright, Fields & Harrison 2000).

Interface of task-based exploration

The current interactive representation allows users to learn the effects of altering the value of a coefficient of a trigonometric function on its curve through performing a task. The interface is illustrated in Figure 3-16

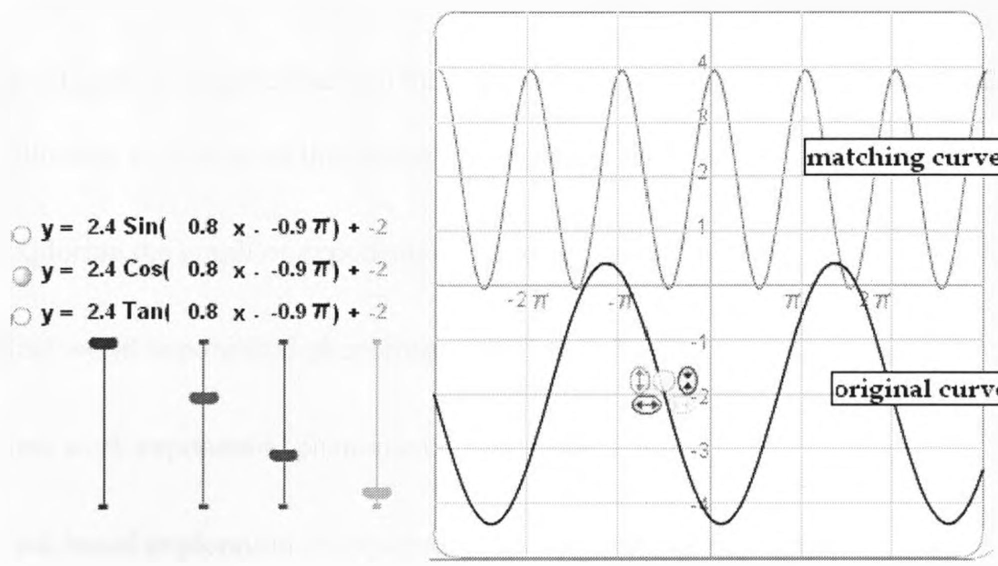

Figure 3-20 interface of the matching task

The task is implemented as matching the original curve with an automatically randomly generated curve. Users can match the curves by interacting via either the handles or sliders. The coefficient values provide quantitative clues about the performed task in the form of substitutive representations, while changing the curve visually provides real time and visual feedback in the form of additive representations.

3.5 Exponential and logarithmic functions

Exponential and logarithmic functions have been used to explain a variety of real world phenomena such as radioactive decay, population growth, and compound interest.

This research explores the following common forms of the exponential and the logarithmic functions.

- $Y = a^{(bx+c)} + d$
- $Y = aLog_b(x+c) + d$

To investigate the exponential and the logarithmic functions, this research considers the following four areas as the primary pedagogical focuses.

- Exploring the graph of exponential and logarithmic functions
- Real world exponential phenomena with lower fidelity
- Real work exponential phenomena with higher fidelity
- Task based exploration of exponential and logarithmic functions

The following sections explain the design and implementation of these concepts.

3.5.1 Exploring the graph

I have considered two ways of interacting with the graphs either via the object manipulation or the concept manipulation techniques. These two ways are intended to allow users to explore the effects of four coefficients of "*a*","*b*","*c*", and "d" on the graph of a trigonometric function.

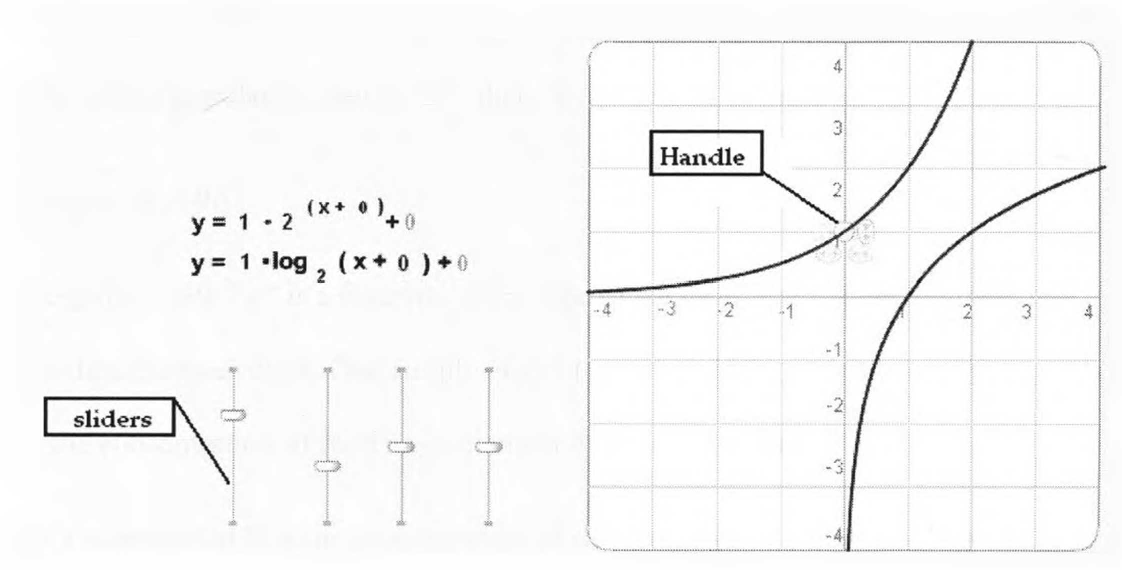

Figure 3-21 interface of the exponential and logarithmic functions

The object manipulation technique can be performed using any of the four provided handles located on the graph. Each handle corresponds to the value of one of the coefficients and can dynamically alter its value.

The concept manipulation technique, however, is facilitated via interacting with sliders, which respectively corresponds to the values of coefficients. As a result, the graph dynamically changes its state and provides visual feedback to the users. Furthermore, both curves of the exponential and the logarithmic functions are drawn on the same grid to support the comparative reasoning.

3.5.2 Model of real world phenomena

Growth of bacteria on a plate is chosen as a real world example of exponential functions. The bacteria's growth of population *"B (t)"* can be explained as an exponential phenomenon that happens within a unit of time *t* with per capita growth rate of *"g"*. This means, in a unit of time, each bacterium on average is able to produce *"g"* children.

If the initial population size is "*B*", then "*b (t)*" can be defined as:

$$B(t) = B.e^{(g.t)}$$

The growth rate *"g"* is a function of the type of bacteria, and the environment that they find themselves in. One simple model to explore is the case where *"g"* depends on the concentration of food as an element of the environment.

Let's assume that R is the concentration of sugar on a plate a model for *"g"* is:

$$g = \frac{2R}{0.3 + R}$$

Increasing the resource concentration *"R"* can amplify the rate of the growth *"g"* and consequently increases the numbers of bacteria over a period of time *"B (t)"*. Also, increasing initial number of bacteria (i.e. the value of parameter *"B"* in the Bacteria equation) in the test tube can amplify the growth of population. However the effect that each of the coefficients of *"B"* or *"g"* has on the overall growth of population is different. This research intends to implement a model to allow users to investigate the effects of those parameters.

Design of the model of bacteria

Two versions of the exponential model of bacteria population is designed and implemented. The first version represents a low fidelity (See Chapter 2) model of the above-mentioned concepts when the shortage of resources is intentionally ignored. In this model bacteria are assumed to reproduce forever, as if there would always be unlimited amount of resources.

The second version, however, represents a more realistic model with a higher level of fidelity (See Chapter 2) to the real world. This model takes into account the fact that after a certain amounts of time the population of bacteria deteriorates as the result of the limitation on the available resources. The pedagogical thought behind the design of these two models is to allow users to gradually develop their understanding of the phenomenon from a lower to a higher fidelity model.

First version of bacteria simulator

In the design of this version, the pedagogical focus is on exploring the effect of the two coefficients of the exponential model. These two coefficients represent the following two elements of a real world environment.

- The concentration of sugar resources
- The initial number of bacteria

The design provides an exploration mechanism by which users make decisions on how to initialize the elements of the environment and investigate them. The investigation is further supported through generating visual feedback and allowing users to explore the consequences of their decisions.

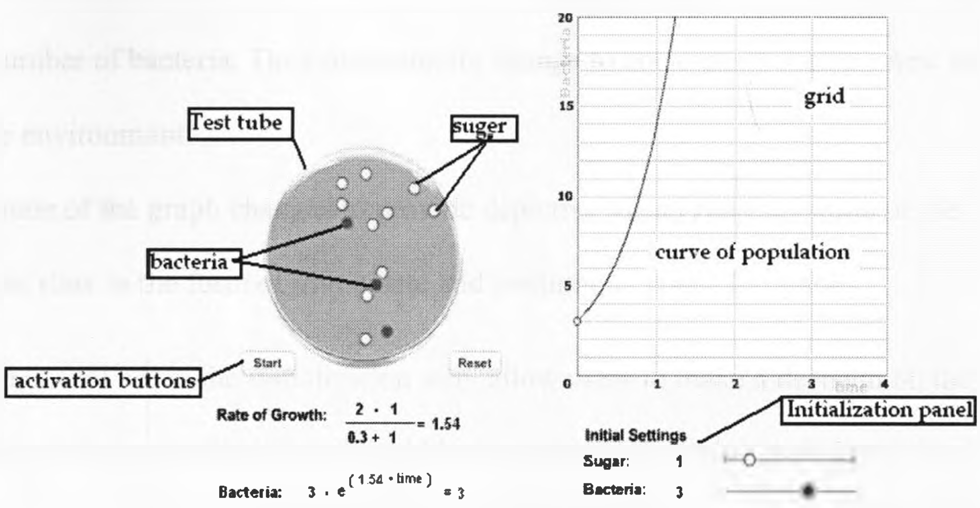

Figure 3-22 simulator of the bacteria growth of population

The interface incorporates three primary visual representations listed as follows:

- The plate of bacteria representing a real world object
- The graph of population growth providing an additive style of representation.
- The formulas and concepts providing a substitutive style of representation

These representations are logically linked together to support the quantitative as well as the qualitative interpretation of underlying concepts.

Setting the environmental elements

Users can interact with the sliders located in the initialization panel to alter the concentration of sugar and change the initial number of bacteria in the test tube. Applying changes to the environment simultaneously alter the state of simulators in three ways listed as follows:

1. The numbers of bacteria and sugars in the test tube change to reflect the new state of the environment.

2. The coefficient values of provided equations represent the rate of the growth and the number of bacteria. They dynamically change to correspond with the new state of the environment.
3. The state of the graph changes to provide depictive visual representation of the current state in the form of immediate and continuous visual feedbacks.

The settings applied, in the initialization step, allow users to make a decision on the state of the environment that they would like to explore. They also provide a concept manipulation mechanism by which users can interact with the actual mathematical formula.

On the other hand, the graph serves as a visual representation, which facilitates mathematical interpretation of the growth of the population. Moreover, it provides a form of before-feedback implying the potential outcome of users' decision (See Chapter 2).

Dynamic representation of growth

After making decisions on the initialization of the environment, supported by the descriptive as well as the depictive visual representations, users can explore the outcome of their decisions and observe how the population of bacteria will increase over a period of time.

To perform these tasks, users can click on the start button and run the simulator. The simulator represents the growth of bacteria in a dynamic and temporal fashion using animation as the interaction type. As the animation starts running, the following tasks are performed simultaneously:

1. In each time frame, new calculated numbers replace the numbers of bacteria in the equation.
2. Numbers of bacteria on the plate are increased to correspond with the values represented in the formula.
3. The state of the curve alters to reflect the current changes.

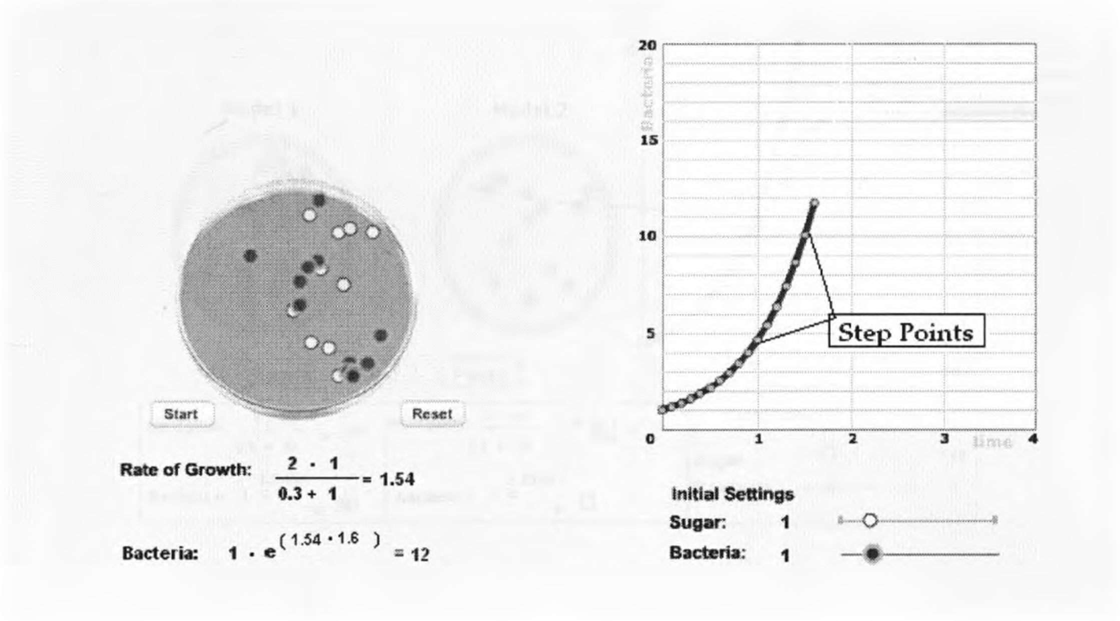

Figure 3-23 step points in the simulator of bacteria growth of population

In addition, at each stage of time a step point as illustrated in the Figure 3-23 is placed at the very end of the graph. The location of each point is calculated and corresponds with its actual number of bacteria. As time passes, the distance between each point and the next one is increased which highlights the exponential property of the growth. The provision of these points is geared toward offloading some of the basic cognitive activity such as tracing and making connections between steps of the transitions. In this sense, they allow users to conceive the state of a temporal phenomenon in a spatial form of transitional steps. Furthermore, it magnifies the exponential property of the phenomenon, allowing users to be clearly exposed to its underlying patterns and logic (See Chapter 2).

Version 2 of the bacteria population

The current version is designed to provide higher fidelity to the real world. Using this version, users can make a comparison between the two represented models which provide two different degrees of fidelity.

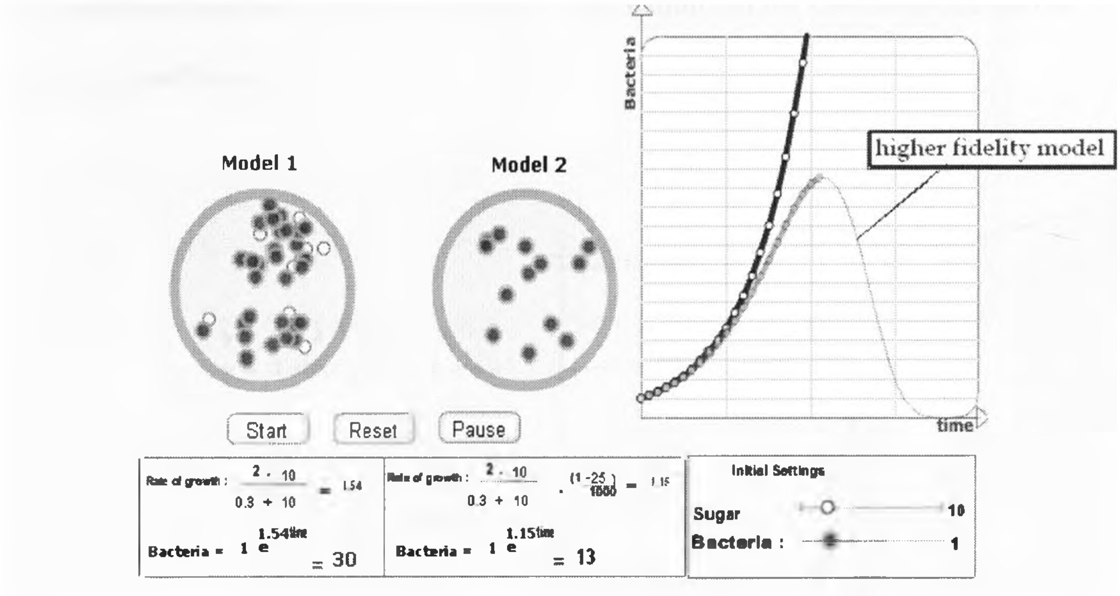

Figure 3-24 higher fidelity implementation of the bacteria growth

In this design, population of bacteria on the plate of the second model (i.e. model 2 in Figure 3-24) will start diminishing at a specified threshold. It illustrates the fact that the amount of sugar resources is limited, and the diminution in the bacteria population represents the shortage in resources. Also, the curve of this model depicts how after passing a period of time the number of bacteria decreases. These situations are illustrated in Figure 3-24.

Placing the two models right beside each other makes it easy for users to investigate their status at any time. Also drawing the graphs of both models on the same curve supports the comparative reasoning (See Chapter 2)

3.5.3 Task-Based learning

Like the task developed for trigonometric functions, I have implemented a matching application for the exponential and the logarithmic function. It is intended to allow users to investigate the effects of the coefficients via performing a matching task. Through this task, users apply the sliders or the handle on the curves to change the values of coefficients.

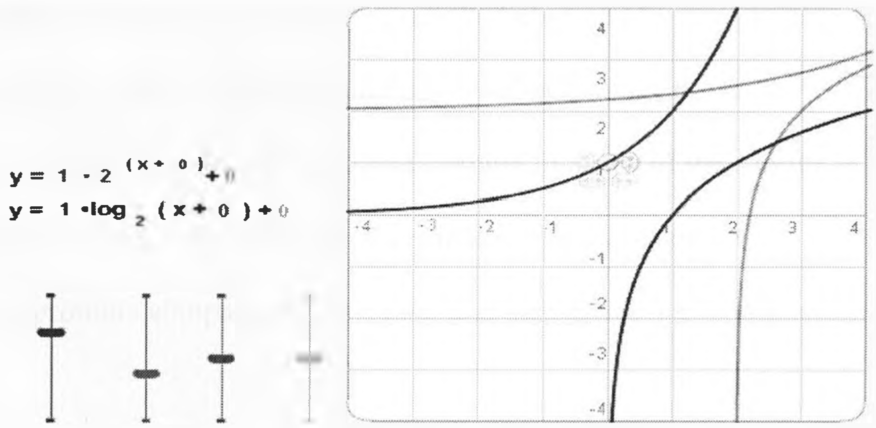

Figure 3-25Matching task for the exponential and logarithmic functions

3.6 Bike motion simulator

The bike simulator targets the primary mathematical concepts of motion such as velocity, distance travel, and acceleration.

This simulator represents a mathematical model of a bike moving from the stopping point, when there are no extra forces such as wind or mechanical friction around.

In this context, the velocity of a bicycle is a function of the gear ratio.

Heuristically, in a high gear one starts more slowly than in low gear, but eventually reaches a higher maximum velocity. The pedagogical focus of this application is to understand the temporal nature of the velocity and to be able to logically link it to the spatial form of a graph.

Interface of bike simulator

The interface of the bike simulator is geared toward providing an exploration environment. It facilitates the users' decision making by allowing them to initialize the parameter of the environment, make sense of mathematical relations, and observe the outcome.

Moreover, it provides a mechanism, which engages users to make sense of different graphs of the motion and utilizes them to interpret the behavior of a moving object. The interpretation is designed to be more qualitative rather than quantitative. Therefore, the simulator hides the mathematical formula in forms of descriptive or substitutive representations. Conversely, it offers more depictive visual representations to promote comparative reasoning and qualitative interpretation (See Chapter 2).

These tasks are facilitated by providing visual representations and a series of interactive components illustrated in Figure 3-26 .

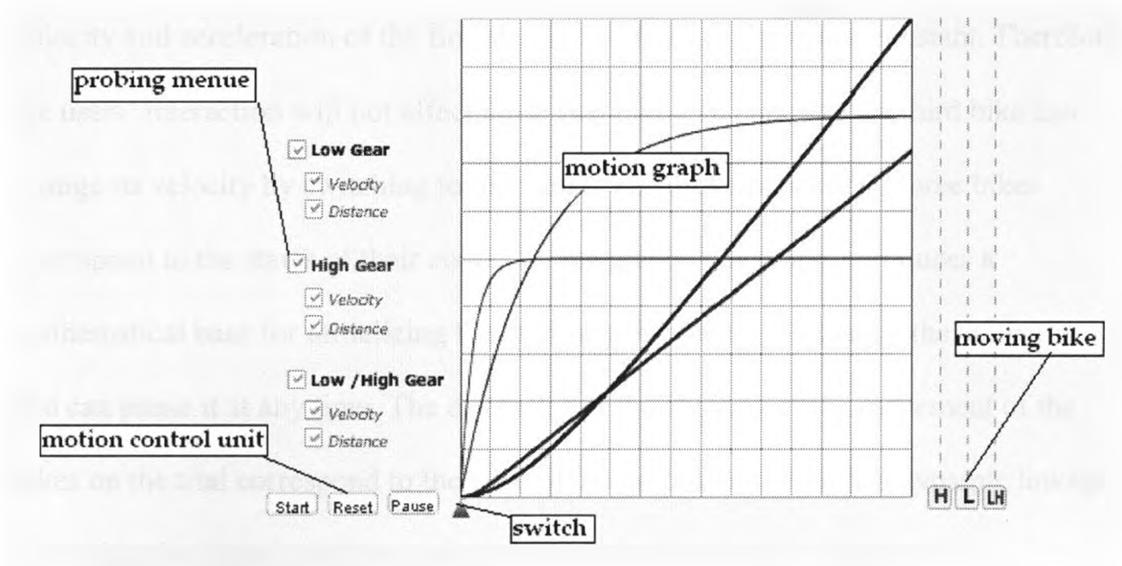

Figure 3-26 interface of the simulator of the bike race

This simulator is composed of the following visual interactive components:

- Moving bike
- Switch
- Probing menu
- Motion graph
- Motion control unit

Moving Bike

The components of the *Moving Bike* are composed of three bikes described as:

- *"H"*, representing a bike in high gear
- *"L"*, representing a bike in low gear
- *"HL"*, representing a bike which starts in low gear but can switch to the high gear
- The trail, representing the distance of travel for each bike

These three bikes are designed to provide an animated form of interaction. The velocity and acceleration of the first two bikes (**H** and **L**) remains constant. Therefore, the users' interaction will not affect those two bikes. In contrast, the third bike can change its velocity by switching to high gear. The movements of all three bikes correspond to the states of their curves on the graph. The graph introduces a mathematical base for initializing the environment. Users can launch the animation and can pause it at any time. The distance and the velocity of the movement of the bikes on the trial correspond to the state of the graphs and provide a dynamic linkage indicating the logical relation among visual components (See Chapter 2).

Switch

The Switch is designed as an interactive slider facilitating a manipulative type of the interaction. The user can drag the switch and place it anywhere on the horizontal axis. The location of the switch on the horizontal axis indicates the time that the bike **LH** switches from low gear to high gear. The range of motion of the slider is considered to remain in the boundary of the horizontal axis.

The switch is deliberately placed on the bottom of the grid as an embedded interactive part of the graph. The proximity of the switch to the grid advocates the grid as the locus of the interaction and as a base for reasoning of the moving bikes.

Probing menu

The menu is designed to allow users to selectively investigate the properties of motion such as velocity and distance. Selecting or deselecting any of the check boxes in the probing menu shows or hides the corresponding graph. The probing menu grants users the flexibility to control the complexity of the representation by adding or removing the corresponding graphs. Also, it makes it possible for users to explore different aspects of the concept from different angles.

Motion graph

The graph represents the curves of the motion for the three bikes. This graph includes the curve of velocity and the curve of distance.

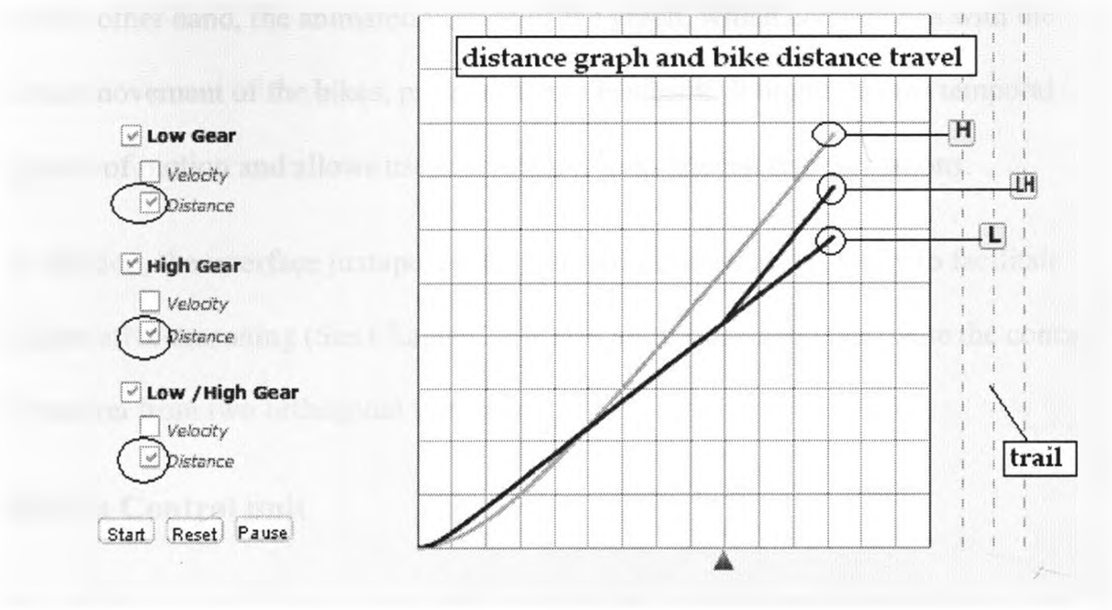

Figure 3-27 the motion graph of the bike simulator

They are intended to facilitate a qualitative interpretation of the bikes' movement. In this sense the state of the distance curve at any given time and the distance that a bike travels on the trail are visually made correlated. Also, the color of curves corresponds to the color of the bikes to provide a visual continuity and draw the users' attention to communicate the inherent relations. Figure 3-27 illustrates how the distance of travel of bikes on the trail corresponds to the graph of distance in terms of location and color.

Graphs are also implemented to serve as visual feedback in two modes of immediate and late feedback. The immediate mode provides a continuous flow of changes to the state of the graph as the result of users interacting with the sliders. At this stage, users can drag and release the slider back and forth and observe potential outcomes in forms of curves of motion. The flexibility of the users in adjusting the location of the switch, as well as the existence of immediate visual feedbacks provided by the simulator encourages the exploration and interpretation of the provided concept.

On the other hand, the animated version of the graph, which corresponds with the actual movement of the bikes, provides a late feedback. It highlights the temporal aspects of motion and allows users to relate it to its special representations.

In addition, the interface juxtaposes the graph of distance and velocity to facilitate comparative reasoning (See Chapter 2). It also allows users to investigate the concept of motion from two orthogonal viewpoints.

Motion Control unit

The *Motion Control* unit is intended to provide an exploration feature allowing users to investigate the outcomes of choosing a specific switching time. It is composed of three buttons of start, reset, and pause. Users can use these buttons to start the movement of the bikes, pause and explore any state of the movement, and reset it. The flexibility in controlling the movement of the bikes and graphs supports the exploratory quality of the simulator by allowing users to process the state of motion at their preferred pace.

The movement of a bike is also paralleled with the dynamic construction of its curve on the graph. Therefore, at any time the user can compare between the state of a bike on the trail and the selected graph of its movement.

How to use the Simulator

Interacting with the bike simulator is designed in two stages of initialization and observation. The initialization is facilitated through interacting with the switch and the probing menu. Users can pick any of the curves listed in the probing menu and change the position of the switch. Consequently, the state of the selected curves alters and provides users with immediate visual feedbacks.

In the exploration stage, users can observe how bikes complete the race. Users can launch the movement and observe how the animated bikes move along the trial and how the graphs of motion evolve respectively.

In addition to providing an exploratory environment, the bike simulator promotes a task-based feature by introducing the goal of wining the race. In this context, users are encouraged to decide on the best switching time that makes the third bike (*"LH"*) to win the race.

To accomplish these tasks users should first explore the states of graphs and patterns of bikes' motion and make sense of their relations.

The state in which the third bike (**LH**) has won the race is represented in Figure 3-28.

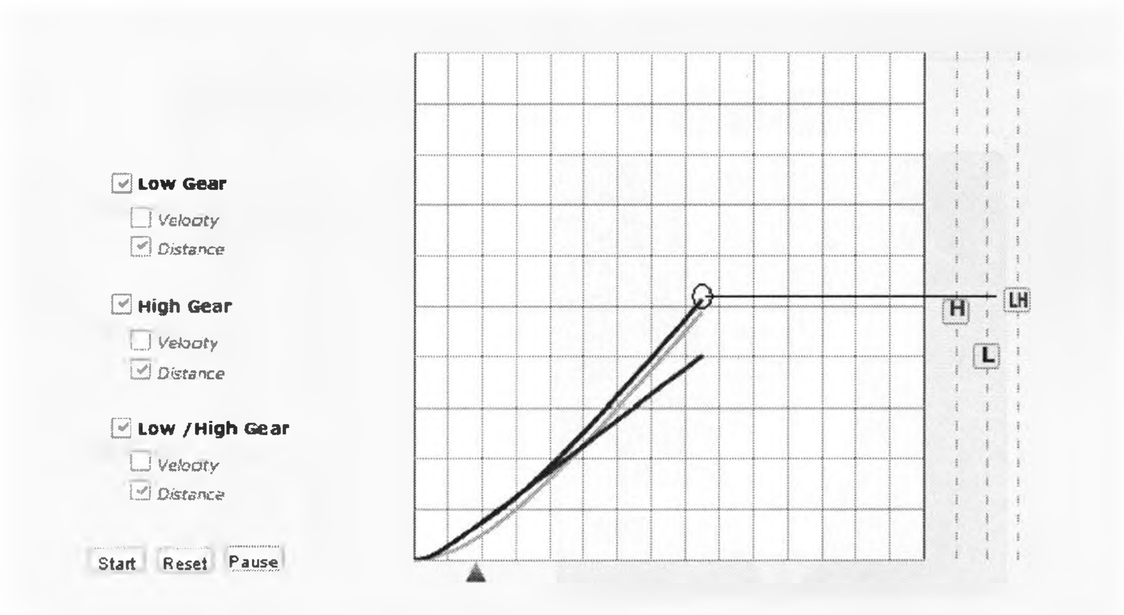

Figure 3-28 the state in which the third bike win the race

3.7 System of inequalities

Inequality interactive is an explorative application, which facilitates the investigation of the mathematical system of inequalities. The presented system of inequalities is composed of three inequalities of two linear and one quadratic inequality. The pedagogical focus of this application is to support users toward identifying the solution region of the system of inequalities on the grid.

Interface

The design of inequality interactive is geared toward providing appropriate visual cues to users allowing them to interpret the solutions of the system in terms of regions of their graphs.

It also provides an interactive mechanism by which users can customize the regions, the equations, and the complexity of the system. The interface is composed of the probing menu, grid, and color-coding hints which are explained in the following sections.

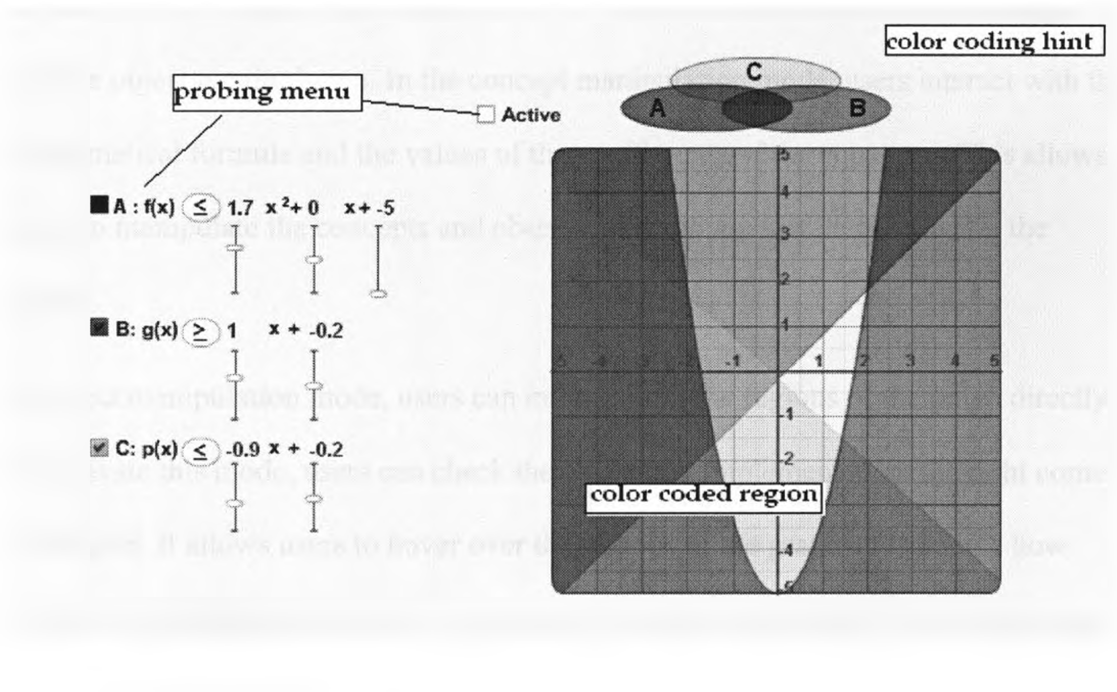

Figure 3-29 the interface of the system of inequalities

3.7.1 Explore regions by probing menu

Users can use the probing menu to add or remove any presented inequalities and therefore customize the complexity of the system. Also, they can alter the state of the sliders to change the values of coefficients of each inequality allowing them to investigate the effects of those coefficients. In addition, the inequality operators are made interactive. Interacting with any of the elements in the probing menu provide visual feedback through changing the solution area on the grid.

3.7.2 Concept manipulation versus object manipulation

This application takes into account two modes of representing the underlying concept. It provides a substitutive as well as an additive form of representations. In the substitutive mode it constantly replaces the values of the coefficients of the inequalities by new ones as the result of users interacting with the sliders. In the additive mode, it continuously changes the state of the regions on the grid to provide immediate feedback and facilitate comparative reasoning.

In addition, the interface provides interactive features in both forms of the concept and the object manipulation. In the concept manipulation mode, users interact with the mathematical formula and the values of the coefficients of the equations. This allows users to manipulate the concepts and observe the outcomes of their action on the graph.

In object manipulation mode, users can interact with the regions of the graph directly. To activate this mode, users can check the check box implemented on the right corner of the grid. It allows users to hover over the regions of the graph and observe how colors of the region and the state of equations change respectively. At any time users can deactivate this mode.

3.7.3 Visual Cues

Color-coding is provided to make the solution of the system more explicit and to provide visual feedbacks to the user. The color of the check boxes beside each inequality function with the color of circles on top of the grid as well as the regions of the graph work in harmony to improve the tracing of the solution and reduce unnecessary demand on the user's cognition (See Chapter 2). Also, superimposing the graphs on each other allows users to makes a comparison among the regions and relates them to the state of the equations. For instance in Figure 3-30 the user has unchecked the last check box and the circle corresponding to that inequality is omitted respectively. Also, the color of the regions of the graph corresponds to this change. The use of colors to categorize and to classify the regions allows users to make a quantitative interpretation (See Chapter 2).

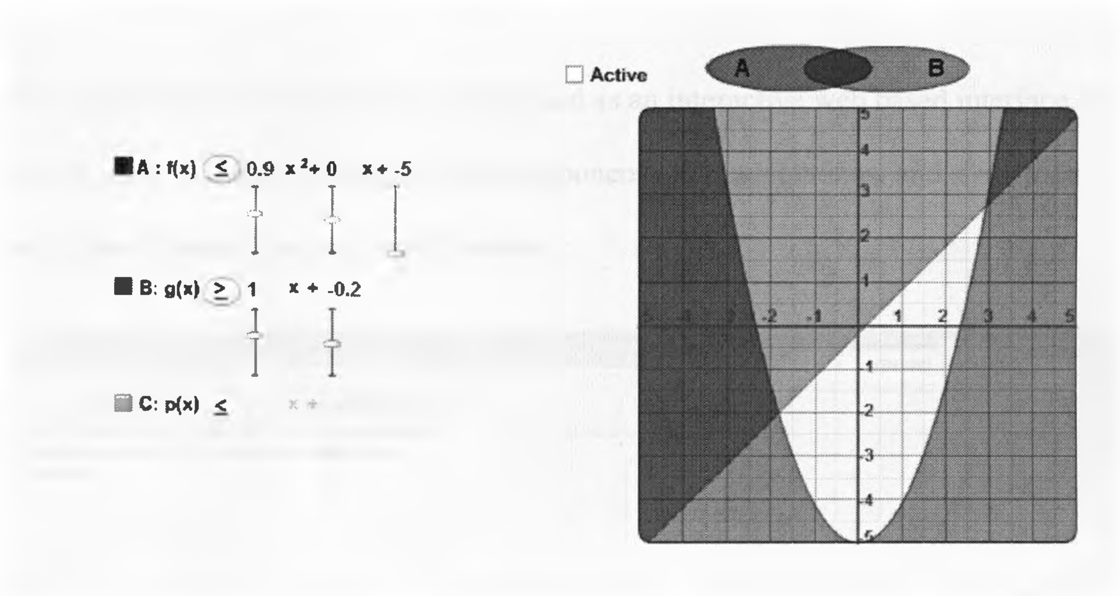

Figure 3-30 the interface of the system of inequality, a specific solution area

Chapter 4
Implementation of the software Architecture

This chapter explains the primary objectives of the design as well as the overall architecture of the two online learning environments developed for this research (See Chapter 3). It also provides a description for some of the classes and objects implemented within those environments. In addition, it discusses major implementation issues that I have encountered through the design process, and it presents my solutions to those issues.

4.1 Design of user Interface

The graphic user interface (GUI) is designed as an interactive web based interface. It allows users to interact with its visual components, add annotations, and share their ideas (see Chapter 3) using a web browser.

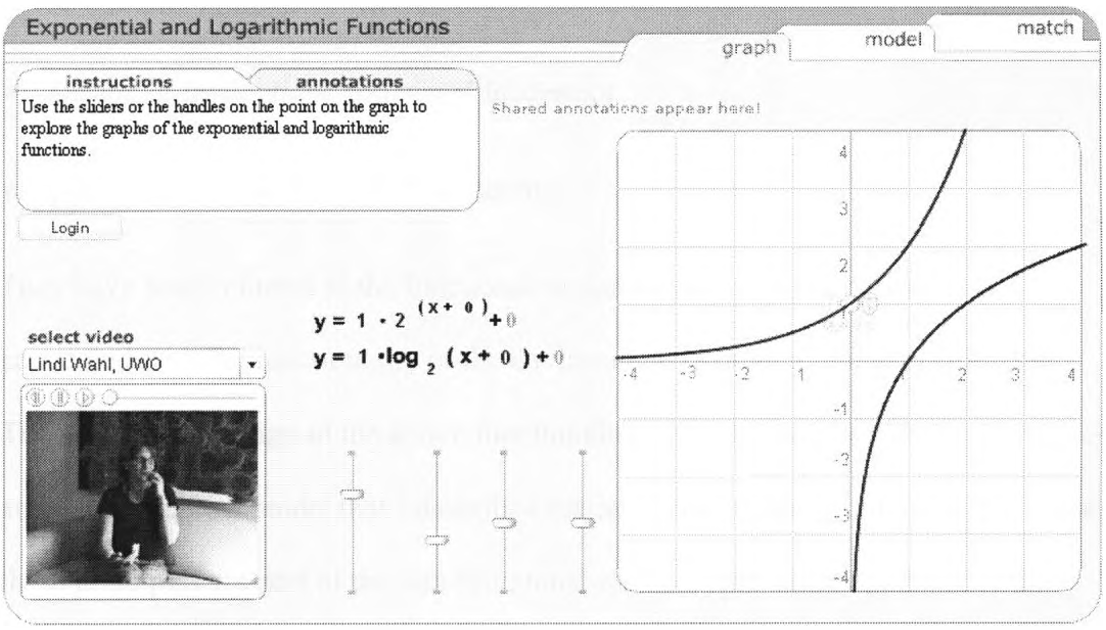

Figure 4-1 interface of the online environment

Moreover, the interface is designed as an integrated environment, which compromises series of interactive mathematical visualizations, videos, instructions, and tools for knowledge sharing. The interface is designed as a template of series of reusable visual components, which can simply plug into any new mathematical supporting tool.

Figure 4-1 illustrates an implemented version of the software. The design is geared toward a client-sever implementation. On the client side the Adobe Flash graphic library and the *Actionscript* programming are utilized to implement the interactive visual representations. Furthermore, the server side takes advantage of Microsoft SQL database and the Active Server Pages (ASP) technology to support the sharing and data retrieving capabilities.

4.1.1 Design objectives and requirements

In the design of the two online learning environments, described in chapter 3, the following three functional requirements are primarily considered.

- Design of multiple visual representations in either a descriptive or a depictive form.
- Design of interactions and visual feedbacks.
- Design of tools which facilitate sharing.

They have been referred to the functional requirements since they are merely concerned with the functionality of the environment (Kotonya, Sommerville 1998). The appropriate design of the above functionalities should comply with the principles and the conceptual model that I described earlier (See Chapter 2 and 3). In that sense, these principles are part of the non-functional requirements or the qualities of the system (Kotonya, Sommerville 1998, Davis 1990).

In addition to the above principles, other non-functional requirements of the system that I took into consideration through the design process are performance, extensibility, maintainability, and testability.

As the environment is a multimedia online application which holds a rich content of texts, videos, and graphics, its performance appears to be a challenging concern. Moreover, since the low performance can considerably affect the user experience, it is critical to make sure that the application maintains a reasonable response time to the users' actions (Nielsen 2000, Nielsen, Loranger 2006).

Additionally, the environment should be flexible enough to be able to easily plug new contents such as new mathematical topics, features, interactive visual components, etc. Flexibility of the system in the future customization is referred to the extensibility (Kotonya, Sommerville 1998, Bass, Clements & Kazman 2003) of the system.

Lastly, testability and maintainability of the system are other concerns in the design of the online environments (see Chapter 3). As those environments, like any other software application, during the implementation process, go through major changes and under rigorous testing, their architecture should be flexible enough to support future maintenances (Kotonya, Sommerville 1998, Bass, Clements & Kazman 2003).

In order to illustrate the components of the architecture, I have employed the graphical notation techniques defined in the UML (Unified Modeling Language) specifications (Larman 1998). The UML diagrams that I have utilized are component diagrams, class diagrams, and sequence diagrams. Next sections describe the architecture of the system as well as its solution to the above mentioned requirements.

4.1.2 Design of the representations

In the design of the visual representations of the two online environments, described in chapter 3, the goal is to provide visual components representing the mathematical concepts. The presented visual components are intended to communicate the underlying mathematical concepts (see Chapter 2 and 3). Moreover, they are designed to provide visual clues allowing the user to be exposed to different points of views of the underlying concepts (see Chapter 2). In this sense, design of these visual representations should comply with several presentation and implementation principles.

As I thoroughly discussed the principles of external representations (see Chapter 2 and 3), in this section, I focus more on the design and implementation issues.

In my design, I have considered providing several representations of the same concept, as it has been studied that they can influence the depth and the quality of learning in different ways (Peterson 1996).

For instance, in the application for the growth of bacteria (See Chapter 2), I have provided three different representations describing the same abstract concept as follows:

- The growth of bacteria in a plate, as a depictive and real world example
- The formula of the population growth, as a descriptive representation
- The graph of the population growth, as a depictive representation

All of the above mentioned representations are informationally equivalent, as they all convey the same idea but in different forms (See Chapter 2). From the OOP perspective (objection oriented programming), I have considered translating the term informationally equivalent to developing a series of objects, which in their own way reflect the same information. As information consists of data and relations among data (see Chapter 2), these objects are required to encapsulate the relevant data and provide methods to define the relations. Therefore, to be informationally equivalent, the same data should be shared among objects, while each object should have its own methods supporting its own way of representing the same data. In this sense, all of the 3 above representations depict the same underlying data but in different forms. Figure 4-2 illustrates how the three different visual representations: the function, the plate, and the graph represent the same underlying data. The underlying data in Figure 4-2 consist of two integers: 3 and 2.6, when 3 is the initial number of bacteria and 2.6 is the concentration of sugar.

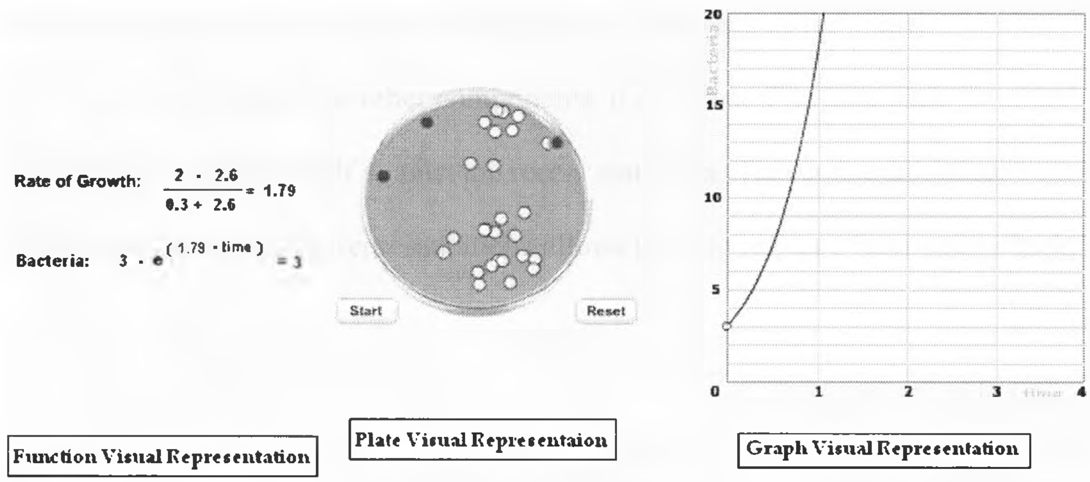

Figure 4-2 informationally equivalent representations

Firstly, the visual representation of the function shows the data (3, 2.6) directly as numbers in the form of the values of coefficients. Secondly, the plate visual representation depicts them as small circles representing sugar and bacteria. Lastly, the graph representation depicts them in the form of an exponential curve of bacteria population growth. The actual data, which constructs the state of each representation, is referred to the state data in this thesis.

On the other hand, when different representations are provided, these representations should establish a logical connection between them to visually provide a harmonic sense (See Chapter 2 and 3). This property is also termed dynamic linkage (Gadanidis, Sedig & Liang 2004). For instance, in the values of coefficients in the formula of population growth change, the other two representations should also be synchronized respectively. To maintain such a synchronization mechanism using object oriented methods, I have considered implementing objects to fulfill that responsibility. The designed objects are required to listen to the events, initiated by the users, and propagate them. The propagation allows the rest of the representations to adjust their state accordingly. These objects are commonly used and are called the listener objects or event handlers (Arnow, Weiss 2000).

The component diagram in Figure 4-3 illustrates how each of the visual representations informs the other components. It also describes how each representation adjusts itself against the recent state data. This communication mechanism among visual representations allows them to reflect a harmonious look.

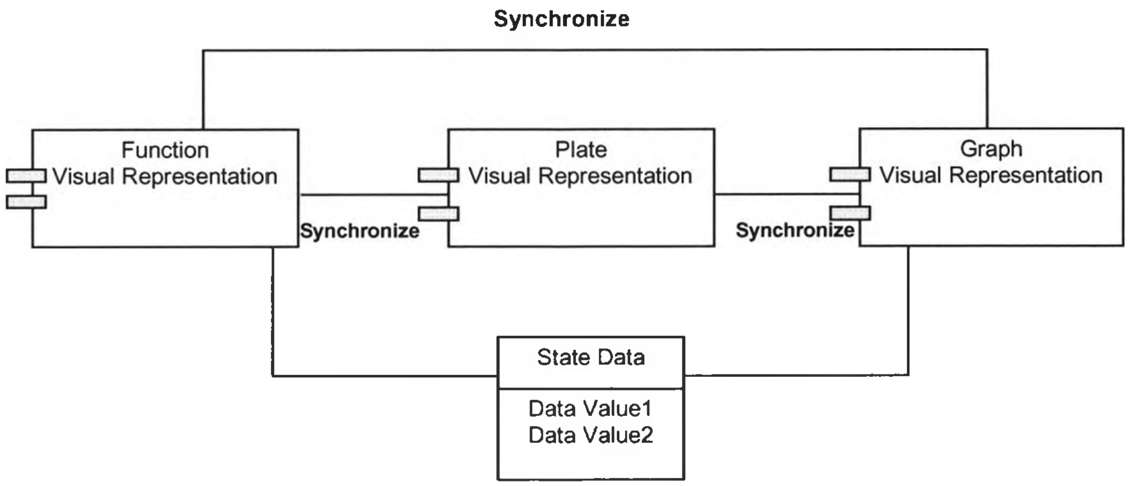

Figure 4-3 the component diagram illustrating the synchronization among visual representations

4.1.3 Design of the interactions

Interaction can be seen as changing the state of visual representations as the result of the user action. To illustrate the process of interacting with a specific representation, a common approach is preparing a series of use-cases or scenarios (Bass, Clements & Kazman 2003). To explain how in my design, a visual representation reacts to the user actions; I have chosen one of the scenarios, which I have prepared. The scenario is chosen from the bacteria growth application described in chapter 3. In that application, users can interact with the sliders to initialize the state of the plate representation, the graph representation, and the formula representation. The following describes the scenario.

- User changes the location of the bar on the slider.
- The slider logic calculates the new location of the bar and propagates it.
- The new propagated data (state data) is accessible by the rest of the representations.
- The slider representation substitutes the old state data with the new one.
- The plate representation alters the numbers of circles (representing sugar) to reflect the new state data.
- The graph representation adjusts the curve to depict the new state data.

As the scenario specifies, I have considered developing a logical unit, responsible for calculating the impact of user action. In this thesis I refer to this logic as interaction logic.

Another essential consideration in designing interactions is the design of object manipulation and concept manipulation interaction techniques. In both cases, users interact with the visual components of the interface. For instance, in the sine curve there are two ways of interacting with the sine curve.

- Via sliders, which change the coefficients of trigonometric functions
- Via a handle located on the graph, which alters the state of the curve

In this sense, there are two interaction points from which users initiate their action. Figure 4-4 illustrates the structure of the interaction points. These points can be implemented either as the object manipulation or the concept manipulation. The object manipulation points are designed as interactive visual components that are embedded within another visual component. For instance, a handle is designed as part of the graph visual representation.

On the other hand, the concept manipulation points are designed as the interactive visual components, which are external to other visual components.

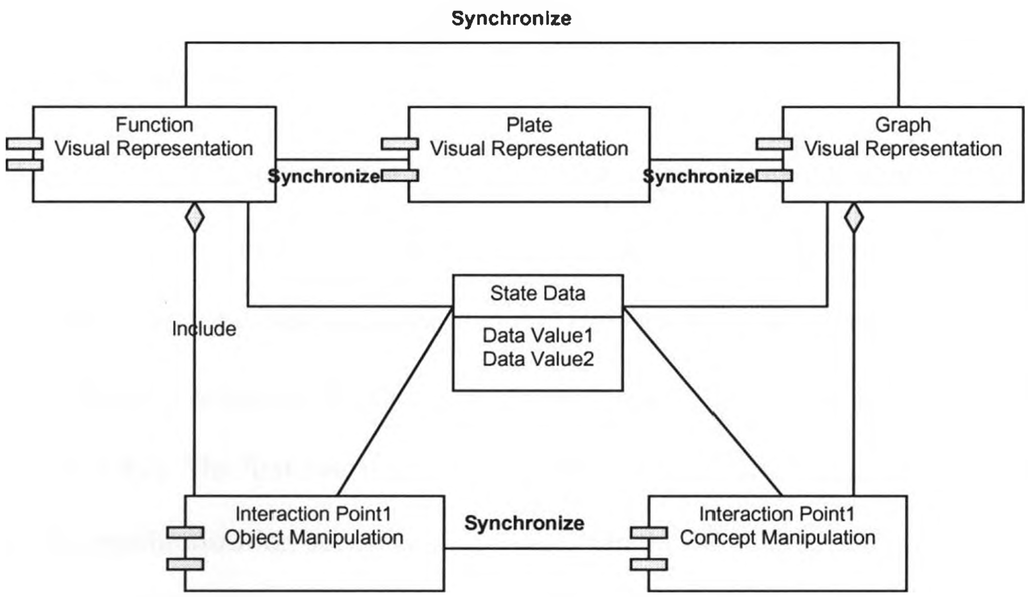

Figure 4-4 component diagram of the interaction and synchronization among visual representations

In both cases the object, which represents a representation, is required to send a message to another object informing it of the occurrence of new changes. This means that the state of these visual components (i.e. slider or handle) as well as the state of the other representations such as graph and pendulum should form a network of representations, which are dynamically linked together. Therefore, no matter on which representation users initiate the interaction, the rest of the representations should reflect the very current status.

Another aspect of the interaction design is implementing the flow of interaction. The flow of the interaction can be continuous by which users get the immediate feedback as the reaction of the system to their action (See Chapter 2). An example of such a situation is when users change the slider and observe that the graph alters instantaneously. It also can be discrete by which the time between the user action and the system reaction is extended allowing users to plan and strategize. This type of flow design is intended to promote decision-making (See Chapter 2). The time of reaction can be either defined automatically by the system (e.g. 10 second after the user action) or initiated and controlled by the user action (e.g. a user pushes a button to see the outcome).

An example of the latter situation is when users interact with the slider on the bike simulator to initialize the switching time of the third bike and then use the start button to observe how the bikes manage to race. The flow of the interaction is represented in the following sequence diagram when users interact with two visual representations (VR1, VR2). The first visual representation provides a time-based discrete feedback as the synchronization accepts a time value. On the other hand, the second visual representation performs the synchronization against the current change of state immediately. It also shows how the other interaction point adjusts its state to be synchronized with the rest of the visual representations.

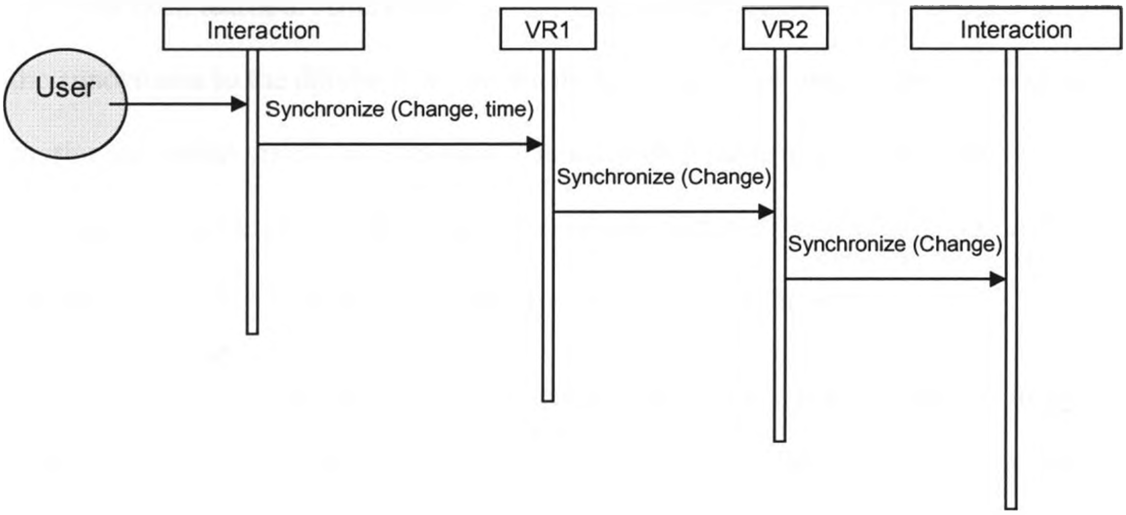

Figure 4-5 the sequence diagram illustrating the steps of user's interaction and synchronization

Also, a primary technical factor in the design of interactive web based applications is their performance. As users' action might be sent to the web server to be processed it will introduce an inherent overhead on the overall performance of the system. This overhead may influence the response time of the system to the user action and generate a poor interactive system. To solve this issue, I have considered designing the interaction logics to be performed in the client side. Therefore users interacting with the visual components all happens on the client side, and the server side is utilized to save and retrieve the state data or to facilitate knowledge sharing.

4.1.4 Design of sharing

Collaboration plays a significant role in building knowledge. As users interact with artifacts to build their own knowledge, integrating sharing capability allows them to enhance and improve their learning process (See Chapter 2).

Along this line of thought and to implement environments in which users can interact with visual representations and share their understandings, I have considered making use of the server side technology and the relational databases. As such, the user interaction occurs in the client side, and then the user annotates and makes notes of what has been learned. Afterward, the user utilizes the web server to transfer and save the annotations to the database. To augment the impact of sharing, I have considered sharing the actual visual representations in addition to sharing the text of annotations. This means users can make different observations and each time capture a snapshot of the observation in the form of a visual representation and share it.

However, making a snapshot by capturing the images of the page in a binary format imposes an extreme inefficiency in the use of resources. Further, it exacerbates the performance of the system by allowing users to save and retrieve immense data. Also, it limits users in making changes or adding further annotations to the snapshots.

My solution to this problem is illustrated in the following scenario:

- The first user makes a change on a visual representation.
- The state of the change is extracted and captured as the state data.
- The sate data is saved in the database.
- The state is ready to be shared as a sharable component.
- The second user requests to have access to the sharable component.
- Sharable component is being retrieved from the database as the state data.

The logic behind each representation is being used to reproduce the presentation out of the state data.

4.2 Design pattern

Design patterns are used to describe the common design problems and to provide solutions for them (Gamma et al. 1995). In the design of the interactive visual components of the online environments, I took advantage of the observer design pattern. This pattern is geared toward providing a "one-to-many dependency among objects". As such, the change of one object automatically affects the rest of the related objects (Gamma et al. 1995). This pattern defines two key objects: the observer and the subject (observable). Observers are basically interested in the changes within the subject (Gamma et al. 1995; Pree 1995). The main responsibility of the subject is to register some observers and to notify them of any changes. On the other hand, observers are responsible to update themselves according to the state that they have been notified of. To apply this pattern in my design, I have considered each of the visual representations to be implemented as an observer and the state data to be employed as the subject. Therefore, any changes to the state data will cause the subject to notify the observers (visual representations), and observers update or adjust their representations accordingly. To further consider the role of the interaction in the applications, I have utilized the MVC (Model-View-Controller) design pattern. MVC extends the observer pattern and defines a new role as the controller. The controller is responsible for interpreting the user action and commanding the state data to change. Later, the change will be propagated among the rest of the visual representations. The next sections illustrate my design and implementation using these patterns.

4.2.1 Observer pattern implementation

I have extended the observer pattern to comply with the requirements of designing the interactive visual representations. In this sense, I have considered defining an independent class representing the state data. The instances of this class propagate the state data as light messages. The observable (subject) is designed to fulfill the following responsibilities:

- Register and un-register related visual representations.
- Get the last state data.
- Notify the registered observers of any changes by propagating the state data.

On the other hand, the responsibility of an observer is to update itself to reflect the current state data. That is, each observer (visual representation) should be able to fulfill this responsibility. To enforce the updating capability of the observers, I have designed the observer as an interface. The primary objective of this interface is to make sure that each class, which implements the interface, is able to update itself according to the state data.

Figure 4-7 shows how these responsibilities are implemented through the methods within two key objects of observer and observable.

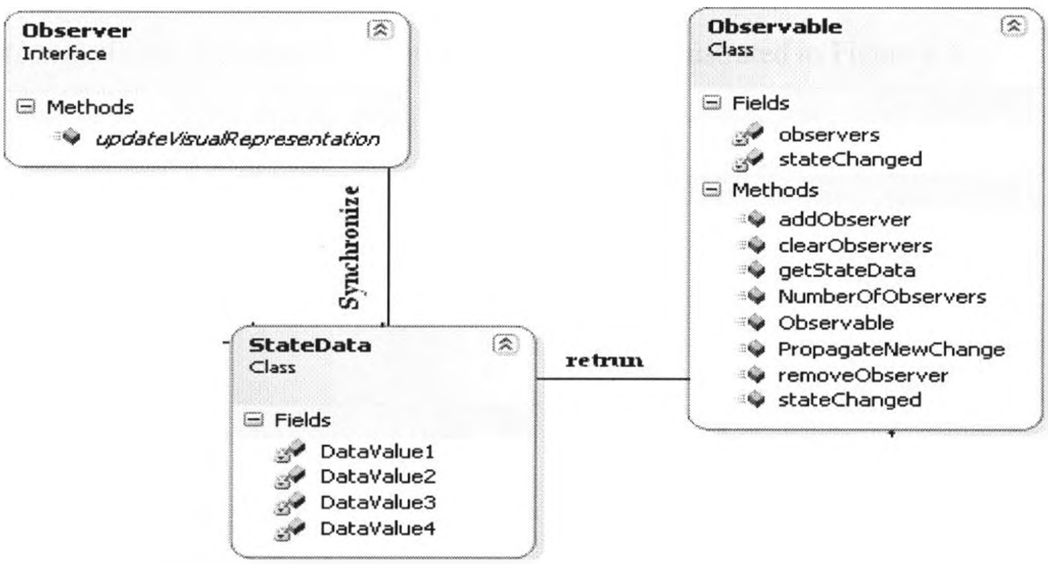

Figure 4-7 the class diagram of the implementation of the observer pattern

4.2.2 Visual Representation implementation

The concrete visual representations such as the plate of bacteria, the graph, and the exponential function of the population growth extend the abstract class of "*AbstractVisualRepresentation*". This class provides a mechanism by which each extended concrete visual representation is responsible to reflect the current state of data. In this sense, an *AbstractVisualRepresentation* implements the observer interface.

On the other hand, each concrete visual representation is designed to provide a mechanism directing the users' action toward the actual interaction logic.

For instance, when the user drags a slider on a visual representation, that visual representation should direct this action toward a specific interaction logic which is responsible for interpreting and assessing this action. In this sense, the *VisualRepresentaion* is designed as an interface to enforce this mechanism.

Therefore, each A*bstractVisualRepresentation* must implement the *VisualRepresentaion* interface. The relation between the *VisualRepresentation, the AbstractVisualRepresentation,* and *the Observer* is illustrated in Figure 4-8

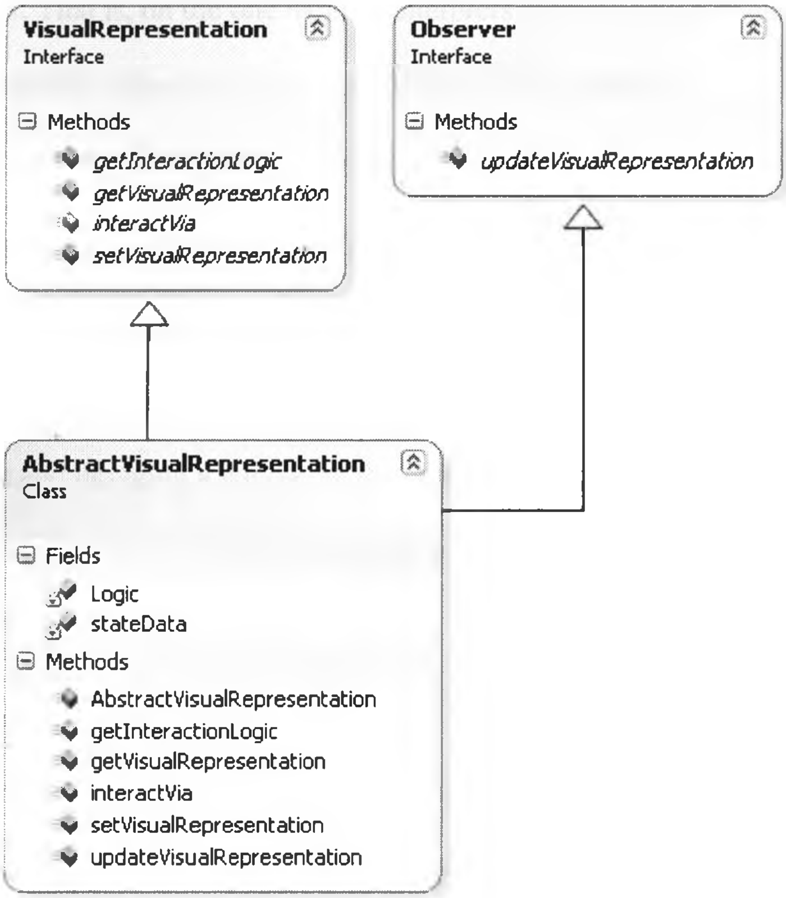

Figure 4-8 class diagram illustrating the implementation of visual representations

Figure 4-8 illustrates how *AbstractVisualRepresentation* implements two interfaces to fulfill the two key responsibilities:

- The synchronization, by implementing *updateVisualRepresentation* () method in the observer interface.

- The interaction, by implementing the *interactionVia*() method in *VisualRepresentation* interface

4.2.3 Interaction Implementation

Concrete interaction points such as sliders or handles implemented in the sine curve application, are designed as visual components. These visual components are backed with some interaction logic. The interaction logic is designed as a semi directional object. That is, on the one hand, it interprets the user action and on the other hand, it manages the appropriate reaction. The interaction logic is designed in compliance with the controller pattern introduced in MVC.

The first direction of the logic, the assessment of the users' action, is closely coupled with the visual representation itself. This means that the user action on different visual representations should be evaluated differently. For instance, dragging the bar on a slider and dragging a handle on the graph are both actions initiated by the user (See Chapter 3). However, calculating the action on the slider and on the graph is different and is tightly bound to separate logic.

The second direction of the interaction logic, the initiation of appropriate reactions, is achieved by changing the state data. As each visual representation contains a built-in synchronization mechanism, it can manage to alter its state to reflect the new committed state data.

These two tasks of assessing the action and providing the reaction are integrated into the *InteractionLogic* interface. This interface makes sure that any class, which implements it, can fulfill the following two responsibilities:

- Interpret the user action in the context of the provoked visual representation.
- Calculate and commit the reaction by changing the state of data.

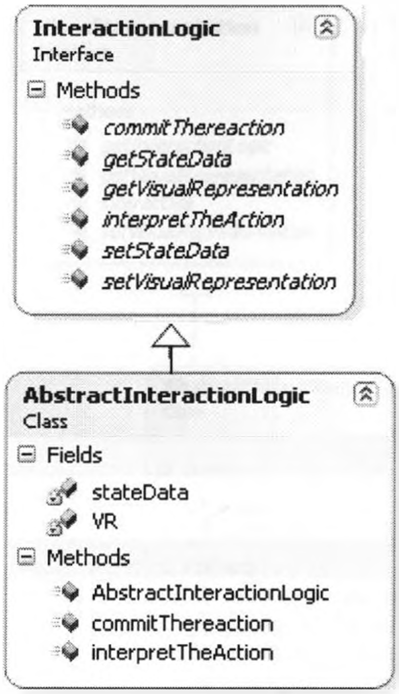

Figure 4-9 class diagram illustrating the implementation of the interaction logic

Figure 4-9 illustrates how the above-mentioned responsibilities are taken into account by implementing the two following functions:

- *interpretTheAction(VisualRepresentation v)*
- *commitTheReaction(StateData data).*

Class diagram

The relation among visual representations, interaction logic, and state data as described in the previous section is illustrated in Figure 4-10.

Each interactive visual representation aggregates interaction logic and a state data. The interaction logic also aggregates the visual representation (VR) to assess the applied action and to commit the necessary changes to the state data. The visual representation adjusts itself based on the new changes to the state data.

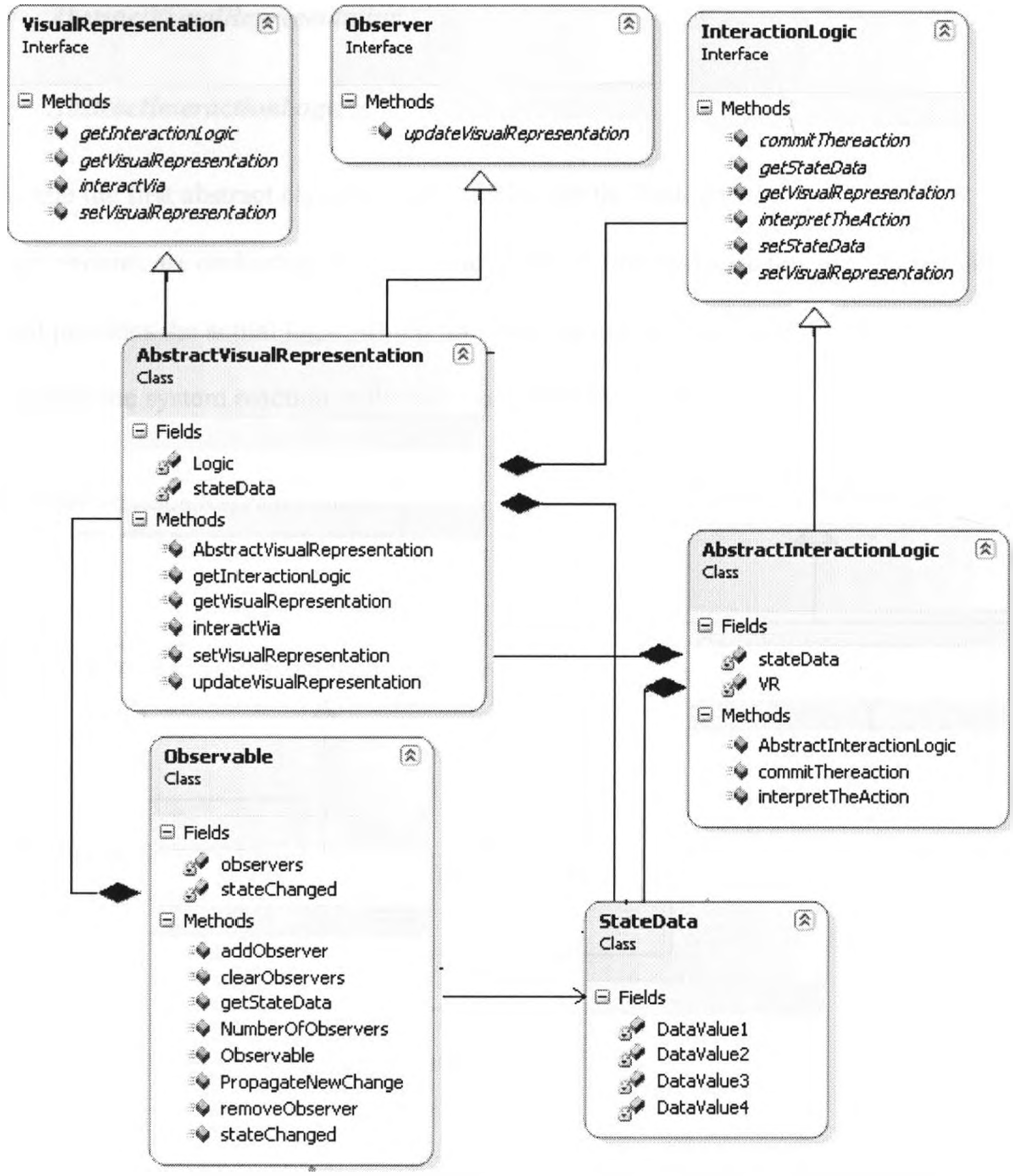

Figure 4-10 class diagram illustrating the interaction, the representation, and the state data

4.3 Visual Components and Interaction

Along the process of implementation for this research, I have developed over 80 classes and more than 15000 lines of codes. The class diagram depicted in Figure 4-11 illustrates the foundational architecture of the software. Other modules of each application extend one of the two following abstract classes:

- *AbstractVisualRepresentation*

- *AbstractInteractionLogic*

While the first abstract class primarily deals with the look and the way visual components are orchestrated, the second abstract class manages the user interaction and provides the actual logic, which translate the user action and provides an appropriate system reaction in the form of visual feedbacks.

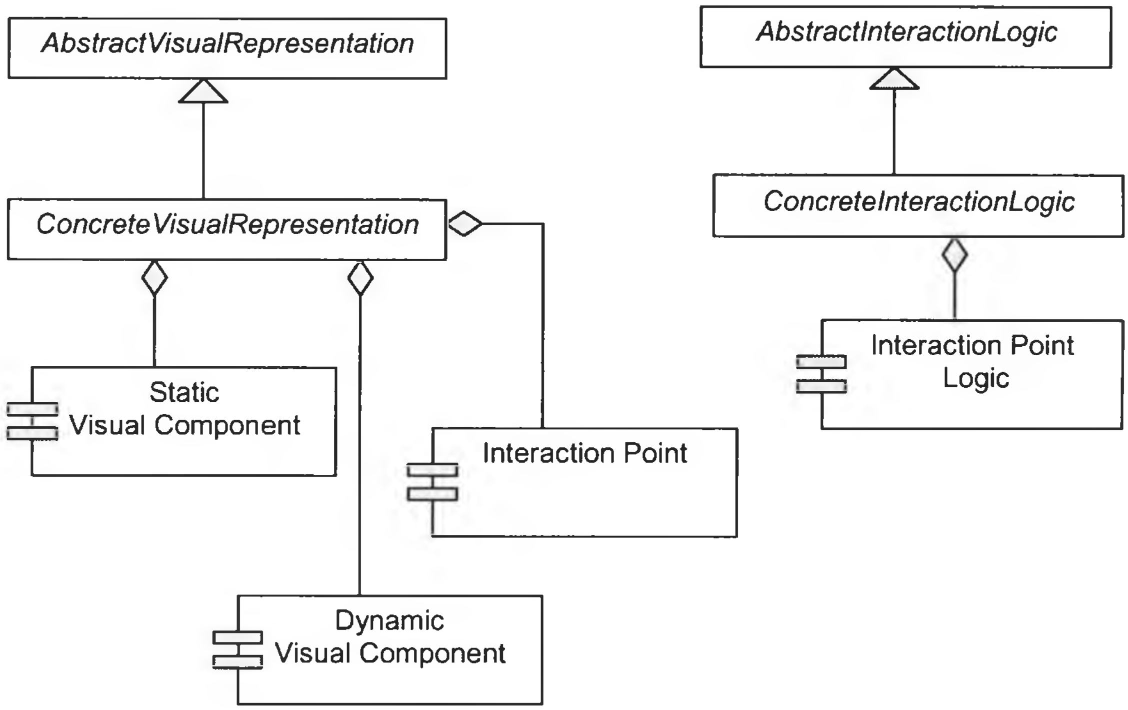

Figure 4-11 class diagram illustrating the hierarchy of visual representations and interactions

Concrete visual representations are designed to include the three following visual components.

- Static visual components

- Dynamic visual components

- Interaction points

Static visual components remain unchanged when a user interacts with them. An example of this type of visual component is the grid in the graph visual representation. In other words these visual components will not change when the underlying state data changes. In contrast, dynamic visual components are designed to respond to changes, which affect the underlying state data. For instance, the curve on the grid continuously changes, when the state data alters.

Interaction points are the visual elements such as sliders and handles, which allow the user to initiate an interaction. Visual representations, which have these interaction points, are made interactive and must implement interaction logic corresponding to the interaction points.

Figure 4-12 shows how concrete visual representations extend the *AbstractVisualRepresentation* and how they are composed of static and dynamic visual components integrated with some interaction points. They also illustrate how the concrete interaction logics are extended from *AbstractInteractionLogic* and correspond to interaction points.

The flow of reaction is illustrated in Figure 4-12. It shows how the user interacts via interaction points and how the interaction logic interprets the action and calculates the reaction. Also, it shows how dynamic components of each visual representation respond to the reaction by performing the synchronization task.

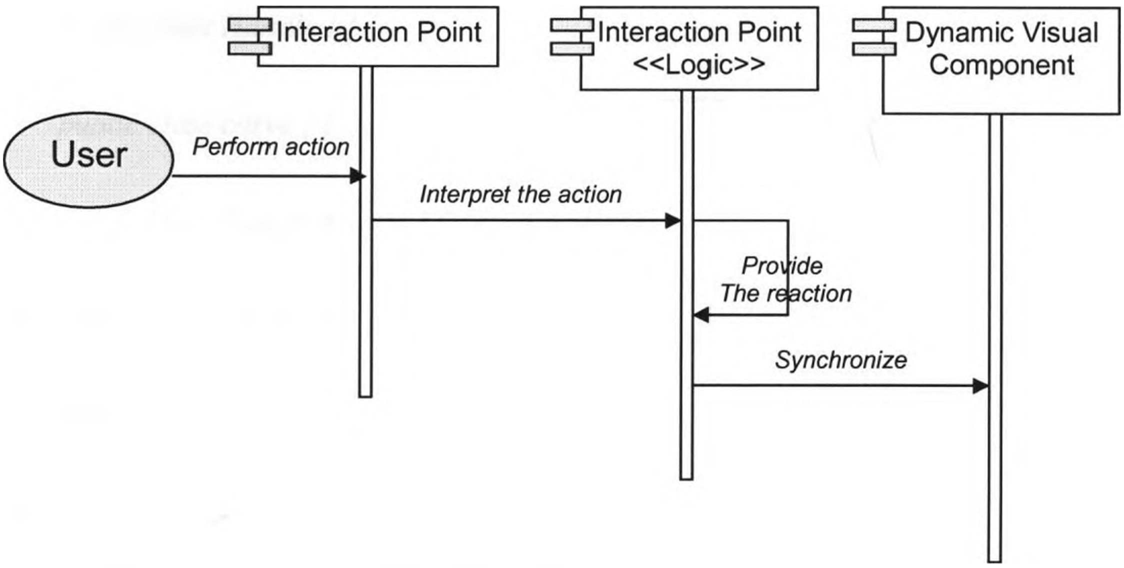

Figure 4-12 the sequence diagram of the interaction

4.3.1 Design of Trigonometric Functions

In the application of trigonometric function the following primary classes are presented and their hierarchy is illustrated in Figure 4-13.

- *public class SliderBar { }*
- *public class Slider { }*
- *public class SliderContainer { }*
- *public class Grid { }*
- *public class Pendelum { }*
- *public class PendelumVR { }*
- *public class GraphTrigonometricVR { }*
- *public class TrigonometricFunctionVR { }*
- *public class TrigonometricMatchVR { }*
- *public class programmablePointsVR{}*

- *public class Handle { }*
- *public class curve { }*
- *public class TangentCurve { }*
- *public class SineCurve { }*
- *public class CosineCurve { }*
- *public class points { }*
- *class annotation { }*

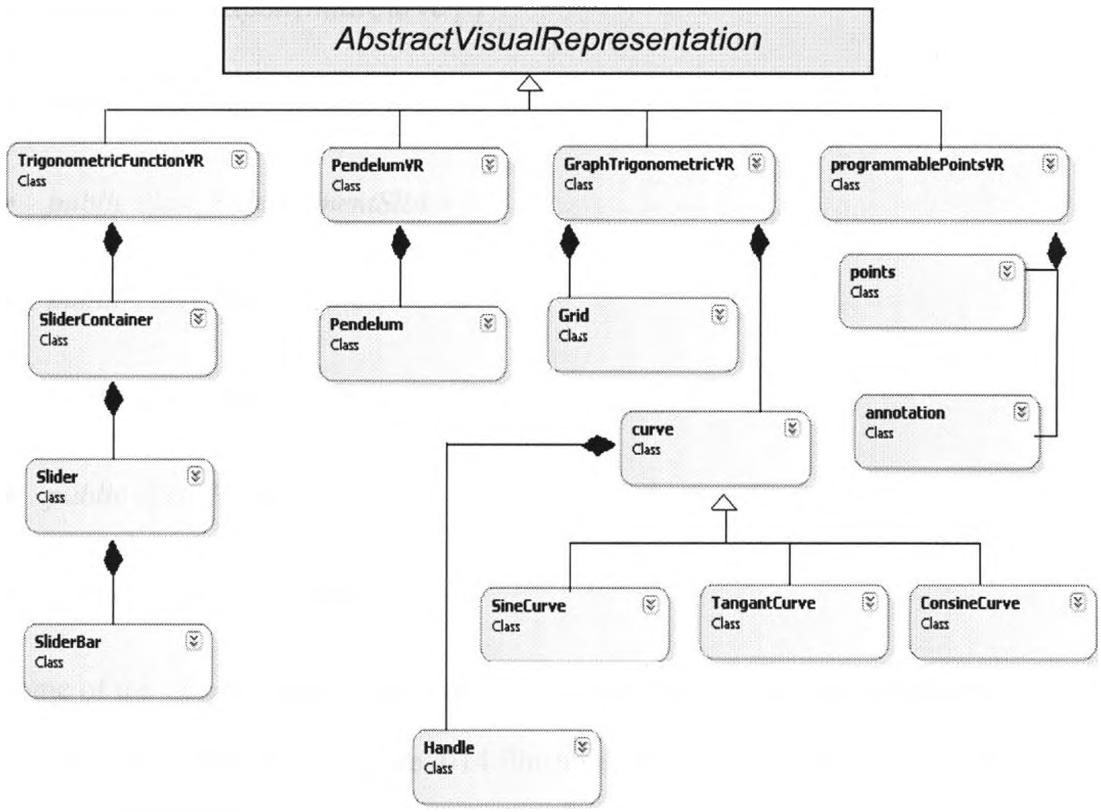

Figure 4-13 the class diagram of the trigonometric functions

As illustrated in the above picture, each visual representation is designed as a concrete visual representation with a prefix of VR, and it is inherited from *AbstractVisualRepresentation*.

4.3.2 Design of exponential and logarithmic functions

Major classes, which are implemented in the design of the exponential and logarithmic functions, are listed as follows:

- *public class ExponentialFunctionVR {};*
- *public class ExponentialGraphVR { };*
- *public class BacteriaPlateVR { }*
- *public class curve { }*
- *public class ExponentialCurve { }*
- *public class Logarithmiccurve{}*
- *public class EnvironmentSlider { }*
- *public class Plate { }*
- *public class Bacteria {}*
- *public class Resources {}*
- *public class ExpHandle{}*

Some of the visual components such as the slider, handle, and annotation are being reused in this application. Figure 4-14 illustrates the hierarchy and the relations among these classes.

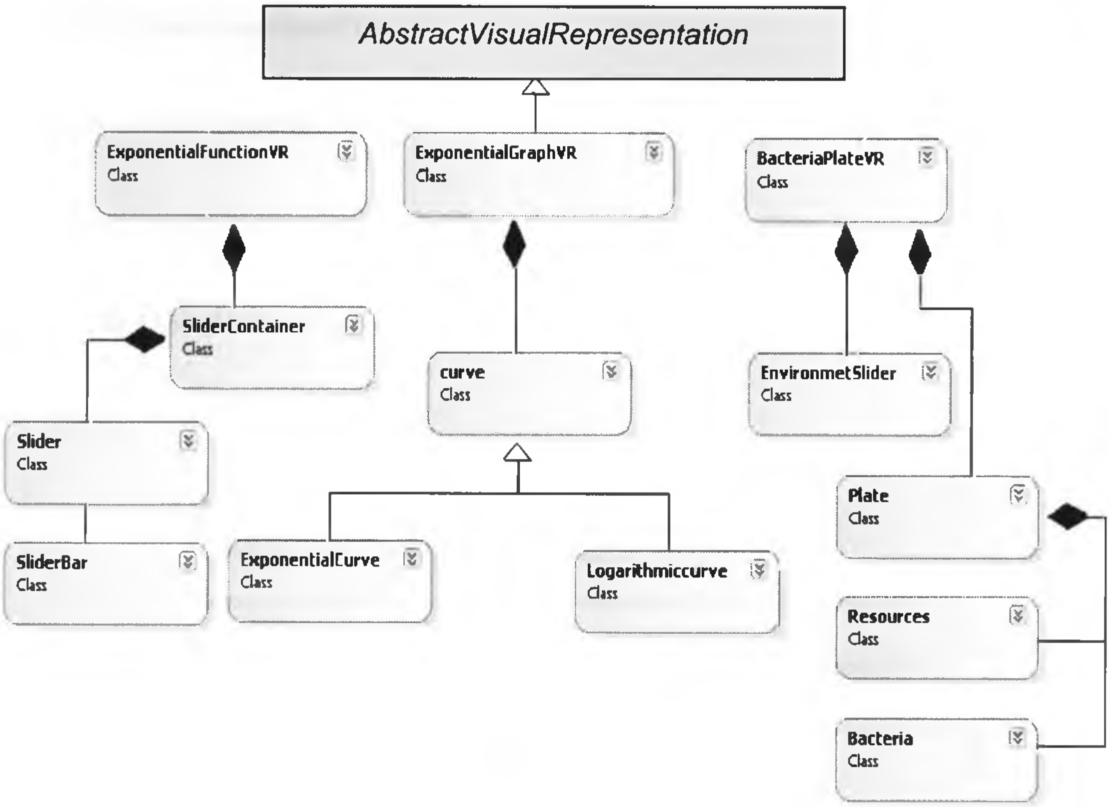

Figure 4-14 the class diagram of the exponential and logarithmic functions

4.3.3 Design of Bike Simulator

- *public class BikeGraphVR { }*
- *public class Curve { }*
- *public class DistanceCurve { }*
- *public class VelocityCurve { }*
- *public class GridBike{}*
- *public class SwitchSlider { }*
- *public class BikersVR { }*
- *public class Bike { }*
- *public class HighGearBike { }*

- *public class LowGearBike { }*
- *public class LowHighBike { }*
- *public class ControlPanelVR { }*
- *public class MotionControlUnit{}*
- *public class ProbingMenu { }*

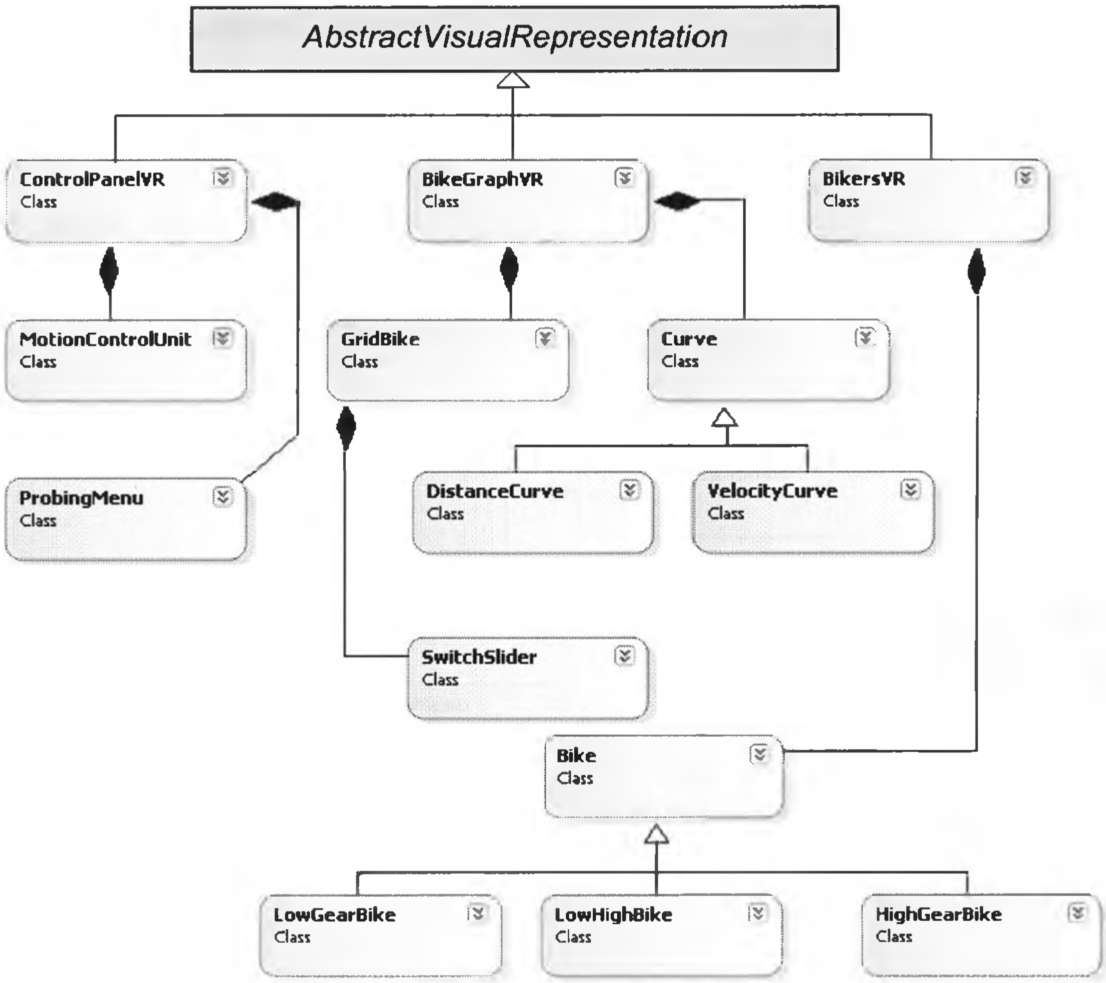

Figure 4-15 the class diagram of the bike simulator

4.3.4 Design of Inequality

- *public class InequalityGraphVR { }*
- *public class Curve { }*

- *public class Grid { }*
- *public class LineCurve { }*
- *public class QuadraticCurve { }*
- *public class EquationVR { }*
- *public class ProbingMenu { }*
- *public class SliderContainer { }*
- *public class Slider { }*
- *public class SliderBar { }*

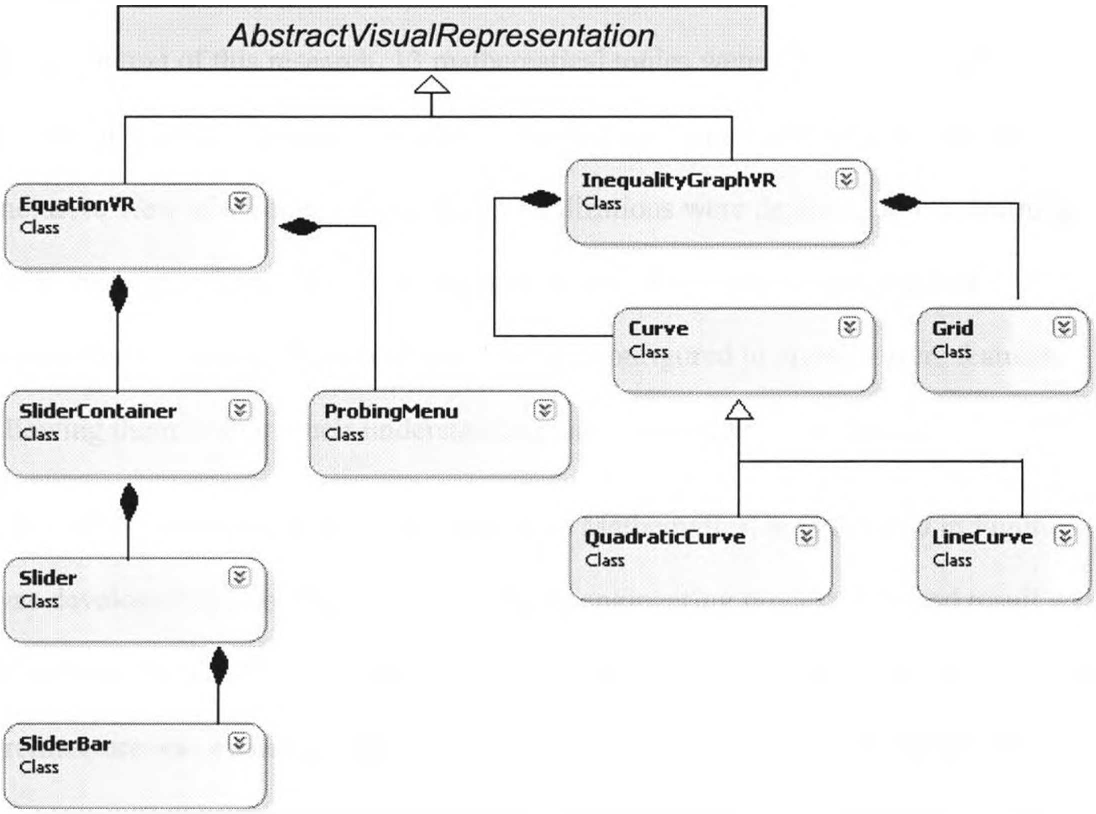

Figure 4-16 the class diagram of the system of inequalities

Chapter 5
Research Summary

This research presents a comprehensive approach in the design of the online mathematics learning environments. It incorporates the principles of human learning with the key concepts of visualization and interaction in order to lay a foundation for the design of effective learning environments. Complying with the foundation, it presents an architectural basis consolidating the theories of learning and the practices of software engineering. Moreover, it introduces a constructivist view of computer-mediated learning environments. That is, the online learning environments are implemented as spaces in which individuals actively participate to build their own knowledge and share it with each other.

In the context of this research, 13 mathematical topics were chosen. For each topic, several interactive visualizations were designed and implemented to provide an inclusive view of the topic. Interactive visualizations were deployed in two learning environments and enriched by instructions and videos. Furthermore, the two environments were made available online and configured to provide users features allowing them to share their understanding and views of presented topics.

The first environment (Digital Windows into Mathematics) as illustrated in Figure 5-1 was developed by adopting an agile software engineering process. The end result was well defined and robust software architecture as described in Chapter 4. The Software architecture was evaluated and reused in the second environment (Mathematics Preparedness Program). The content of this environment is illustrated in Figure 5-2.

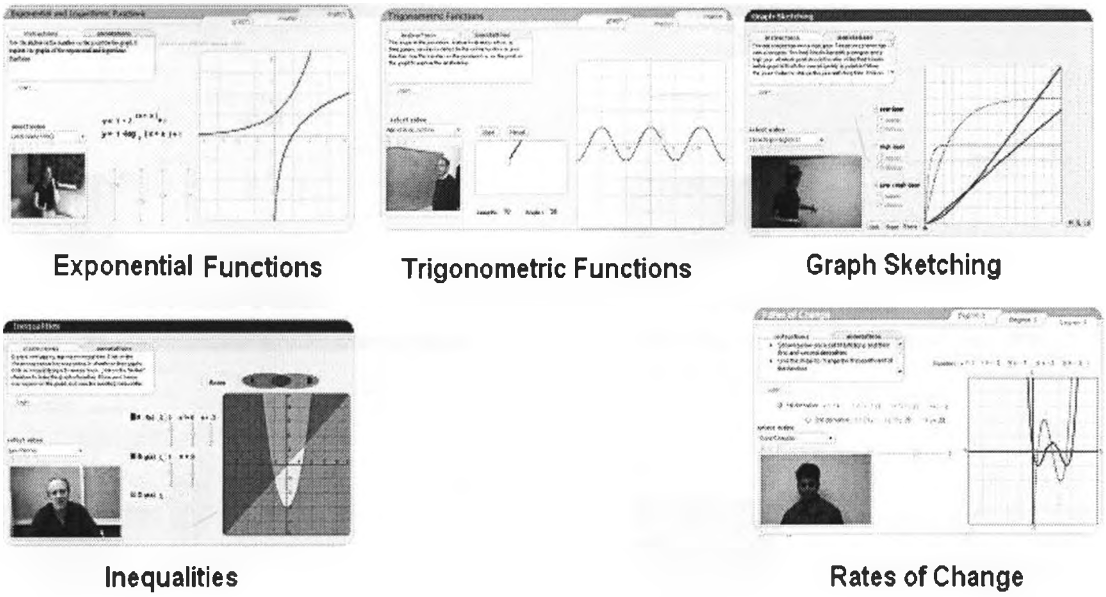

Figure 5-1 Digital Windows into Mathematics

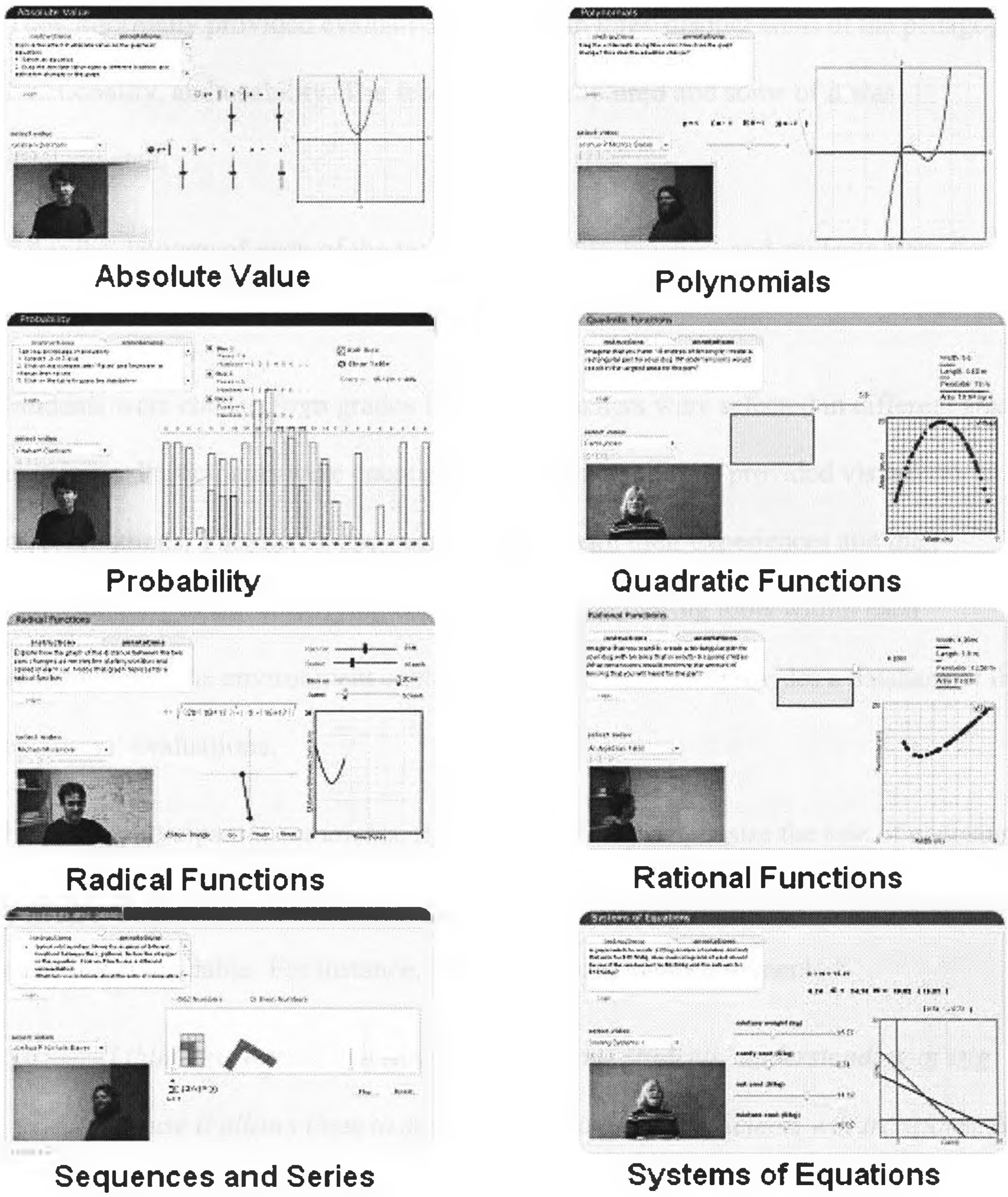

Figure 5-2 the Mathematics Preparedness Program

5.1 Evaluation and Feedback

Along with the process of development, a group of educators and usability experts from Huron-Perth Catholic District School Board, London District Catholic School Board, and University of Western Ontario's Department of Education have rigorously tested the two environments.

They informally provided evaluative feedback in three distinct areas of the pedagogy, functionality, and usability. The feedback was captured and some of it was implemented.

After the delivery of each of the two environments, teachers and students from the two school boards assessed the end products.

Students were chosen from grades 11-12 and teachers were selected in different areas in science. Participants were encouraged to interact with the provided visual representations. They have also been asked to share their experiences and their understanding of the underlying concepts using the sharing tools within each environment. The environment captures the users' idea and provides a database of the end users' evaluations.

From the participant's comments, it appears that they emphasize the role of choosing appropriate interactive and visual representations which make the underlying concepts more understandable. For instance, one of the participants commented:

> *"I think this interactive would help improve students' understanding of trig because it allows them to see how changing the coefficients will transform the graph without the tedious work of drawing the graphs by hand"*

Similarly, another participant pointed out the appropriate use of color and the dynamic linkage between visual representations allowing the underlying concepts to stand out.

> *"This worked very well, I liked that the sliders were in the same order as the parameters and the colors matched. It is very easy to see the connection between what is happening in the exponential and logarithmic functions, as well as the transformations happening"*

Identifying a similar issue, on the inequality application, another participant notes:

> *"Great interactive, students get a sense of what an inequality is. I particularly like the different colors to identify the regions"*

On the other hand, participants have approved the clarifying role of providing the real life applications in terms of making sense of the presented concepts. In that regard, one of the participants stated:

> *"The real life application shows students that trig functions can be used to model something they may see as useful or at the very least realistic"*

Also, one of the teachers commented:

> *"Fantastic! I would have students use this activity before using real pendulums. This could be the initial demonstration and give them an idea of what they could expect to see."*

Moreover, most participants found the videos very illustrative. The following are some of the comments made in that regard:

> *"I found the videos to be very clarifying in terms of the growth rate equation and the relationship between the sugar and bacteria levels"*

Another participant mentioned:

> *"I liked this model! It is easy to use and easy to see the exponential growth of bacteria. It also ties in well with the Variable growth rate video"*

Furthermore, participants stressed that the use of annotation, as well as other interactions, improved the student's engagement. The following is one of the participant's comments:

> *"Overall, I really like these activities. I think they will help engage students and help students understand the connection between exponential and logarithmic relations and how they are used."*

In addition, the two environments and their interactive visualizations are published on a publicly available Website for use by other educational institutions and for access for research purposes. Currently more than 10,000 teachers and students have visited those Websites and annotated their understandings.

5.2 Future work

A study of formal evolution of the two environments is left for future work. This thesis presents an informal analysis of design and pedagogy to show the potential of these environments.

From the design point of view, the first environment (Digital Windows into Mathematics) took account of the feedbacks provided by usability experts and incorporated the methods of software architecture (see Chapter 4). The developed solution is designed and implemented based on flexible software architecture (see Chapter 4). This flexibility allows developers to apply changes rapidly without affecting the whole application. The second environment was developed based on the same architecture and proved the flexibility of the architecture in terms of reusing components and objects as well as reducing the development time.

From the pedagogical perspective, the effectiveness of the environments and their interactive visual components were evaluated by teachers and students. The teachers and students' feedback illustrate the compliance of the two environments with the objective of supporting an active way of learning.

Although each domain introduces its own topics and content, this research provides a systematic approach in terms of the design of software products addressing the pedagogical concerns and mapping them into visual contents.

Bibliography

Aring, rhem, P. & Liljenström, H. 1997, "On the Coevolution of Cognition and Consciousness", *Journal of Theoretical Biology,* vol. 187, no. 4, pp. 601-612.

Ackoff, R.L. 1999, *Ackoff's Best: His Classic Writings on Management, P4,* John Wiley & Sons, Inc, New York.

Ackoff, R.L. 1989, "*From Data to Wisdom*", *Journal of Applied Systems Analysis,* vol. 16, pp. 3-9.

Adelson, B. & Soloway, E. 1985, "The Role of Domain Experience in Software Design", *IEEE Trans.Softw.Eng.,* vol. 11, no. 11, pp. 1351-1360.

Ainsworth, S. 1999, "The functions of multiple representations", *Computers and Education,* vol. 33, no. 2-3, pp. 131-152.

Alessi, S.M., Stanley, M. & Alessi, S.M. 2001a, *Multimedia for learning: methods and development,* 3rd edn, Allyn and Bacon, Boston.

Alessi, S.M., Stanley, M. & Alessi, S.M. 2001b, *Multimedia for learning: methods and development,* 3rd edn, Allyn and Bacon, Boston.

Andriessen, J., Baker, M. & Suthers, D. 2003, *Arguing to learn: confronting cognitions in computer- supported collaborative learning environments,* Dordrecht ;.

Arnow, D.M. & Weiss, G. 2000, *Introduction to programming using Java: an object-oriented approach,* Java 2 update edn, Addison-Wesley, Reading, Mass.

Atkinson, R.C. & Shiffrin, R.M. 1968, "Human Memory: A Proposed System and its Control Processes" in *Psychology of Learning and Motivation*, ed. Kenneth W. Spence and Janet Taylor Spence, Academic Press, , pp. 89-195.

Baker, E.A.(. 2000, "CASE-BASED LEARNING THEORY: IMPLICATIONS FOR SOFTWARE DESIGN", *Journal of Technology and Teacher Education,* vol. 8, no. 2, pp. 85-95.

Barker, P. 1994, "*Designing interactive learning*
" in *Design and production of multimedia and simulation-based learning material*, eds. T.D. Jong & L. Sarti, Kluwer, .

Bass, L., Clements, P. & Kazman, R. 2003, *Software architecture in practice,* 2nd edn, Addison-Wesley, Reading, Mass.

Becerra-Fernandez, I. & Sabherwal, R. 2001, "Organizational knowledge management: A contingency perspective", *Journal of Management Information Systems,* vol. 18, no. 1, pp. 23.

Blyth, A. 1999, "Using Stakeholders, Domain Knowledge, and Responsibilities to Specify Information Systems' Requirements", *Journal of Organizational Computing and Electronic Commerce,* vol. 9, no. 4, pp. 287.

Brey, P. 2005, "The Epistemology and Ontology of Human-Computer Interaction", *Minds and Machines,* vol. 15, no. 3-4, pp. 383-398.

Cannon-Bowers, J.A., Salas, E. & Converse, S. 1993, "*Shared mental models in expert team decision making*" in *Individual and group decision making: current issues*, ed. N.J. Castellan, L. Erlbaum, Hillsdale, N.J., pp. 221-247.

Chandrasekharan, S. & Stewart, T.C. 2007, "The Origin of Epistemic Structures and Proto-Representations", *Adaptive Behavior,* vol. 15, no. 3, pp. 329-353.

Chang, K.E., Chen, Y.L., Lin, H.Y. & Sung, Y.T. 2008, "Effects of learning support in simulation-based physics learning", *Computers & Education,* vol. 51, no. 4, pp. 1486-1498.

Chinnappan, M. 1998, "Schemas and mental models in geometry problem solving", *Educational Studies in Mathematics,* vol. 36, no. 3, pp. 201-217.

Clark, R.E. 1983, "Reconsidering Research on Learning from Media", *Review of Educational Research,* vol. 53, no. 4, pp. 445-459.

Clark, R.C. & Mayer, R.E. 2003, *E-Learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning,* 1st ed, Jossey-Bass/Pfeiffer, San Francisco, CA.

Coll, R.K. & Treagust, D.F. 2003, "Investigation of secondary school, undergraduate, and graduate learners' mental models of ionic bonding", *Journal of Research in Science Teaching,* vol. 40, no. 5, pp. 464-486.

Colom, R., Rebollo, I., Palacios, A., Juan-Espinosa, M. & Kyllonen, P.C. 2004, "Working memory is (almost) perfectly predicted by *g*", *Intelligence,* vol. 32, no. 3, pp. 277-296.

Corbett, A.T., Anderson, J.R. & Anderson, J.R. 2001, "Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes", *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, New York, NY, USA, pp. 245.

Cowan, N. 2005, *Working Memory Capacity,* Psychology Press.

Craik, K.J.W. 1952, *The nature of explanation,* The University press, Cambridge Eng.

Davenport, T.H. & Grover, V. 2001, "General Perspectives on Knowledge Management: Fostering a Research Agenda", *Journal of Management Information Systems,* vol. 18, no. 1, pp. 5-21.

Davenport, T.H. & Prusak, L. 1998, *Working knowledge: how organizations manage what they know,* Harvard Business School Press, Boston, Mass.

Davis, A.M. 1990, *Software requirements: analysis and specification,* Prentice Hall, Englewood Cliffs, N.J.

De Jong, T. & Van Joolingen, W.R. 1998, "Scientific Discovery Learning with Computer Simulations of Conceptual Domains", *Review of Educational Research,* vol. 68, no. 2, pp. 179-201.

Devlin, K.J. 2003, *Mathematics, the science of patterns: the search for order in life, mind, and the universe,* 1st Owl books edn, Henry Holt, New York.

Doyle, J.K. & Ford, D.N. 1998, "Mental models concepts for system dynamics research", *System Dynamics Review,* vol. 14, no. 1, pp. 3-29.

Duffy, T.M. & Knuth, R.A. 1990, "*Hypermedia and instruction. Where is the match?*" in *Designing hypermedia for learning*, eds. D.H. Jonassen & H. Mandl, Springer-Verlag, Berlin ;; New York, pp. 199-225.

Durrance, B. 1998, "Some explicit thoughts on tacit learning", *Training & Development,* vol. 52, no. 12, pp. 24-29.

Foltz, C. & Overton, W.F. 1995, "Proof Construction: Adolescent Development from Inductive to Deductive Problem-Solving Strategies", *Journal of Experimental Child Psychology,* vol. 59, no. 2, pp. 179-195.

Gadanidis, G., Sedig, K. & Liang, H. 2004, "Designing online mathematical investigation", *Journal of Computers in Mathematics and Science Teaching,* vol. 23, no. 3, pp. 275-298.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. 1995, *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison -Wesley.

Gazzaniga, M.S. 1988, *Perspectives in memory research,* MIT Press.

Gray, B. 1989, *Collaborating: finding common ground for multiparty problems,* 1st edn, Jossey-Bass, San Francisco.

Henninger, S. 1995, "Developing domain knowledge through the reuse of project experiences", *SIGSOFT Softw.Eng.Notes,* vol. 20, no. SI, pp. 186-195.

Hermian, J., Tabachneck-Schijf, M. & Simon, H.A. 1996, "Alternative Representations of Instructional Material" in *Forms of Representation*, D. Peterson edn, Intellect Books, Oxford, UK, pp. 28-45.

Hutchins, E.L., Hollan, J.D. & Norman, D.A. 1987, "Direct manipulation interfaces (excerpt)", pp. 468-470.

Hutchins, E. 1990, "The technology of team navigation", , pp. 191-220.

Jacobson, M.J. & Spiro, R.J. 1995, "Hypertext learning environments, cognitive flexibility, and the transfer of complex knowledge: An empirical investigation. Journal of Educational Computing Research", *Journal of Educational Computing Research,* vol. 12, no. 4, pp. 301-333.

JOHNSON, A.M. 1990, "Speed of mental rotation as a function of problem solving strategies", *Perceptual and motor skills,* vol. 71, no. 1, pp. 803-806.

Johnson-Laird, P.N. 1983, *Mental models: towards a cognitive science of language, inference, and consciousness,* Harvard University Press, Cambridge, Mass.

Jonassen, D.H. 1990, "*Semantic network elicitation: Tools for structuring hypertext.*" in *Hypertext: state of the art*, eds. R. McAleese & C. Green, Intellect Ltd.; Oxford, England.

Keith, S.T. 2004, "Oversimplification of 'the truth' can act as a barrier to learning", *Physics Education,* vol. 39, no. 6, pp. 461-462.

Keller, J.M. 1983, "*Motivational design of instruction*" in *Instructional-design theories and models :an overview of their current status*, ed. C.M. Reigeluth, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 383-434.

Kirschner, P.A. 2002, "Cognitive load theory: implications of cognitive load theory on the design of learning", *Learning and Instruction,* vol. 12, no. 1, pp. 1-10.

Kirsh, D. & Maglio, P. 1994, "On distinguishing epistemic from pragmatic action", *Cognitive Science,* vol. 18, pp. 513-549.

Kirsh, D. 1997, "Interactivity and multimedia interfaces", *Instructional Science,* vol. 25, no. 2, pp. 79-96.

Kosslyn, S.M. 1994, *Image and brain: the resolution of the imagery debate,* MIT Press, Cambridge, Mass.

Kotonya, G. & Sommerville, I. 1998, *Requirements engineering: processes and techniques,* John Wiley, Chichester; New York.

Kozma, R.B. 1991, "Learning with Media", *Review of Educational Research,* vol. 61, no. 2, pp. 179-211.

Kyllonen, P. & Christal R. 1990, "Reasoning ability is (little more than) working memory capacity?!" *Intelligence,* vol. 14, pp. 389-433.

Lally, V. & Laat, M.D. 2002, "Cracking the code: learning to collaborate and collaborating to learn in a networked environment [Electronic version", *In G. Stahl (Ed.), Proceedings of Computer Support for Collaborative Learning*Erlbaum, , pp. 160.

Lam, A. 2000, "Tacit Knowledge, Organizational Learning and Societal Institutions: An Integrated Framework", *Organization Studies,* vol. 21, no. 3, pp. 487-513.

Larkin, J. & Simon, H.A. 1987, "Why a diagram is (sometimes) worth ten thousand words", *Cognitive Science,* vol. 11, pp. 65-99.

Larman, C. 1998, *Applying UML and patterns: an introduction to object- oriented analysis and design,* Prentice Hall PTR, Upper Saddle River, N.J.

Luger, G.F. 1994, *Cognitive science: the science of intelligent systems,* Academic Press, San Diego.

Maglio, P.P., Wenger, M.J. & Copeland, A.M. 2008, "Evidence for the role of self-priming in epistemic action: Expertise and the effective use of memory", *Acta Psychologica,* vol. 127, no. 1, pp. 72-88.

Martin, E.E. & Rainey, L. 1993, "Student achievement and attitude in a satellite delivered high school science course", *The American Journal of Distance Education,* , no. 7, pp. 1-54.

Mayer, R.E. 2001, *Multimedia learning,* Cambridge University Press, Cambridge, U.K.; New York.

Mohammed, S. & Dumville, B.C. 2001, "Team mental models in a team knowledge framework: Expanding theory and measurement across disciplinary boundaries", *Journal of Organizational Behavior,* vol. 22, no. 2, pp. 89.

Mulder, I., Swaak, J. & Kessels, J. 2002, "Assessing Group Learning and Shared Understanding in Technology-Mediated Interaction", *Educational Technology & Society,* vol. 5, no. 1, pp. 35-47.

Nanda, P., Liang, H. & Sedig, K. 2005, "Interaction and Interactivity in Online Mathematical Applets: Two Sides of the Same Coin", *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*, eds. P. Kommers & G. Richards, AACE, Montreal, Canada, pp. 1284.

NCTM 2006, *Principles and Standards for School Mathematics,* Reston, VA, National Council of Teachers of Mathematics, Inc.

Nielsen, J. 2000, *Designing Web usability,* New Riders, Indianapolis, Ind.

Nielsen, J. & Loranger, H. 2006, *Prioritizing Web usability,* New Riders, Berkeley, Calif.

Nonaka, I. & Konno, N. 1999, "The Concept of "Ba": Building a Foundation for Knowledge Creation" in *Knowledge management yearbook: 1999-2000*, eds. J.W. Cortada & J.A. Woods, Butterworth-Heinemann, Boston, pp. 37-51.

Nonaka, I. 1994, "A Dynamic Theory of Organizational Knowledge Creation", *Organization Science,* vol. 5, no. 1, pp. 14-37.

Norman, D.A. 1991, "Cognitive artifacts" in *Designing interaction: psychology at the human-computer interface*, ed. J.M. Carroll, Cambridge University Press, Cambridge England; New York.

Norman, D.A. 1993, *Things that make us smart: defending human attributes in the age of the machine,* Addison-Wesley Pub. Co., Reading, Mass.

Norman, D.A. 1988, *The psychology of everyday things,* Basic Books, New York.

Norman, D.A. 1983, "Some observations on mental models" in *Mental models*, eds. D. Gentner & A.L. Stevens, L. Erlbaum Associates, Hillsdale, N.J., pp. 7-15.

Norman, D.A. & Norman, D.A. 1990, *The design of everyday things,* 1st Doublay/Currency edn, Doubleday, New York.

Nurmi, R. 1999, "Knowledge-Intensive Firms" in *Knowledge management yearbook: 1999-2000*, eds. J.W. Cortada & J.A. Woods, Butterworth-Heinemann, Boston, pp. 168-179.

Ormrod, J.E. 2007, *Human Learning,* 5th edn, Prentice Hall.

Paas, F., Tuovinen, J.E., Tabbers, H. & Van Gerven, P.W.M. 2003, "Cognitive Load Measurement as a Means to Advance Cognitive Load Theory", *Educational Psychologist,* vol. 38, no. 1, pp. 63.

Parnafes, O. & Disessa, A. 2004, "Relations between Types of Reasoning and Computational Representations", *International Journal of Computers for Mathematical Learning,* vol. 9, no. 3, pp. 251-280.

Peterson, D. 1996, *Forms of representation: an interdisciplinary theme for cognitive science,* Intellect, Oxford.

Pree, W. 1995, *Design patterns for object-oriented software development,* Addison-Wesley Pub. Co., Wokingham, England; Reading, Mass.

Prusak, L. 1999, "What's up with knowledge management?" in *The knowledge management yearbook, 1999-2000*, eds. J.W. Cortada & J.A. Woods, Butterworth-Heinemann, Boston, pp. 3-7.

Qin, Y. & Simon, H.A. 1992, "Imagery and mental models in problem solving" in *American Association for Artificial Intelligence Spring Symposium Series* Lawrence Erlbaum Associates, Bloomington, Indiana, pp. 18-23.

Rogers, Y., Rutherford, A. & Bibby, P.A. 1992, *Models in the mind: theory, perspective and application,* Academic Press, London; San Diego.

Roschelle, J. & Teasley, S.D. 1995, "Construction of shared knowledge in collaborative problem solving" in *Computer-supported collaborative learning,* O'Malley, C. edn, Springer-Verlag., New York, NY, pp. 67-197.

Rosson, M.B. & Gold, E. 1989, "Problem-solution mapping in object-oriented design", *OOPSLA '89: Conference proceedings on Object-oriented programming systems, languages and applications*ACM, New York, NY, USA, pp. 7.

Ruddle, R.A., Howes, A., Payne, S.J. & Jones, D.M. 2000, "The effects of hyperlinks on navigation in virtual environments", *International Journal of Human-Computer Studies,* vol. 53, no. 4, pp. 551-581.

Schein, E.H. 1999, *The corporate culture survival guide: sense and nonsense about culture change,* 1st edn, Jossey-Bass, San Francisco, Calif.

Schnotz, W. 2002, "Towards an Integrated View of Learning from Text and Visual Displays", *Educational Psychology Review,* vol. 14, no. 1, pp. 101-120.

Sedig, K., Rowhani, S., Morey, J. & Lian, H. 2003, "Application of information visualization techniques to the design of a mathematical mindtool: a usability study,", *Palgrave Macmillan Ltd.,* , pp. 142-152.

Sedig, K., Rowhani, S. & Liang, H.N. 2005, "Designing interfaces that support formation of cognitive maps of transitional processes: an empirical study", *Interacting with Computers,* vol. 17, no. 4, pp. 419-452.

Sedig, K. 2008, "From Play to Thoughtful Learning: A Design Strategy to Engage Children With Mathematical Representations", *Journal of Computers in Mathematics and Science Teaching,* vol. 27, no. 1, pp. 65-101.

Sedig, K., Klawe, M. & Westrom, M. 2001, "Role of interface manipulation style and scaffolding on cognition and concept learning in learnware", *ACM Trans.Comput.-Hum.Interact.,* vol. 8, no. 1, pp. 34-59.

Sedig, K. & Liang, H. 2007, "On the design of interactive visual representations: Fitness of interaction", *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007*, eds. C. Montgomerie & J. Seale, AACE, Vancouver, Canada, pp. 999.

Sedig, K. & Liang, H. 2006, "Interactivity of Visual Mathematical Representations: Factors Affecting Learning and Cognitive Processes", *Journal of Interactive Learning Research,* vol. 17, no. 2, pp. 179-212.

Sedig, K. & Sumner, M. 2006, "Characterizing Interaction with Visual Mathematical Representations", *International Journal of Computers for Mathematical Learning,* vol. 11, no. 1, pp. 1-55.

Sherry, D. 2006, "Mathematical reasoning: induction, deduction and beyond", *Studies in History and Philosophy of Science,* vol. 37, no. 3, pp. 489-504.

Shneiderman, B. & Wald, C. 1987, *Designing the user interface: strategies for effective human-computer interaction,* Addison-Wesley, Reading, Mass. ;; Don Mills, Ont.

Simon, H.A. 1996, *The sciences of the artificial,* 3rd edn, M.I.T. Press, Cambridge.

Skemp, R.R. 1979, *Intelligence, learning, and action: a foundation for theory and practice in education,* Wiley, Chichester Eng.; Toronto.

Sowa, J.F. 1984, *Conceptual structures: information processing in mind and machine,* Addison-Wesley, Reading, Mass.; Don Mills, Ont.

Spector, J.M. 1995, "Integrating and Humanizing the Process of Automating Instructional Design", *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES,* vol. 140.

Spence, R. 2001, *Information visualization,* Elsevier.

Spence, R. 1999, "A framework for navigation", *International Journal of Human-Computer Studies,* vol. 51, no. 5, pp. 919-945.

Steen, L.A. 1988, "The Science of Patterns", *Science,* vol. 240, no. 4852, pp. 611-616.

Sveiby, K.E. 1999, "Tacit knowledge" in *Knowledge management yearbook: 1999-2000*, eds. J.W. Cortada & J.A. Woods, Butterworth-Heinemann, Boston, pp. 18-27.

Tapscott, D. 1997, "Growing Up Digital: The Rise of the Net Generation", McGraw-Hill, Inc. New York, NY, USA

Teasley, S.D. & Roschelle, J. 1993, "Constructing a joint problem space: The computer as a tool for sharing knowledge" in *Computers as Cognitive Tools*, Lajoie,S. P. ;Derry,S. J. edn, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 229-258.

Tufte, E.R. 2000, *Visual explanations :images and quantities, evidence and narrative,* Fourth printing, with revisions, April 2000 edn, Graphics Press, Cheshire, Conn.

Tulving, E. 1983, *Elements of episodic memory,* Clarendon Press ;, Oxford Oxfordshire.

Urbanek, S. and Unwin, A. R. 2001, Making trees interactive – KLIMT. In "*Proc. of the 33th Symposium of the Interface of Computing Science and Statistics*".

Vandervert, L.R. 2003, "How working memory and cognitive modeling functions of the cerebellum contribute to discoveries in mathematics", *New Ideas in Psychology,* vol. 21, no. 1, pp. 15-29.

Wagener, H. & Strothotte, T. 1998, *Computational Visualization: Graphics, Abstraction, and Interactivity,* Springer-Verlag New York, Inc, Secaucus, NJ, USA.

Ware, C. 2000, *Information visualization: perception for design,* Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

Wright, P., Fields, B. & Harrison, M. 2000, "Analysing Human-Computer Interaction as Distributed Cognition: The Resources Model", *Human Computer Interaction,* vol. 15, pp. 1-42.

Zeleny, M. 1987, "Management Support Systems: Towards Integrated Knowledge Management", *Human Systems Management,* vol. 7, no. 1, pp. 59.

Zhang, J. Norman, D. A. 1994, "
Representations in Distributed Cognitive Tasks
", *COGNITIVE SCIENCE,* vol. 18, no. 1, pp. 87-122.

Zhang, J. 1997, "The Nature of External Representations in Problem Solving", *Cognitive Science,* vol. 21, no. 2, pp. 179-217.