2005

The feedback model in peer-to-peer based active e-learning

I-Ching Fong
San Jose State University

THE FEEDBACK MODEL

IN

PEER-TO-PEER BASED ACTIVE E-LEARNING

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

I-Ching Fong

August 2005

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

Dr. Xiao Su

Dr. Donald Hung

Dr. Rod Fatoohi

APPROVED FOR THE UNIVERSITY

7/19/05

ABSTRACT

THE FEEDBACK MODEL IN PEER-TO-PEER BASED ACTIVE E-LEARNING

by I-Ching Fong

E-learning has utilized the power of the Internet in education for many years now. However, the current client-server model in e-learning lacks an effective feedback system that is one of the essential elements in any successful educational model. The crude feedback in this client-server model focuses primarily on providing users with the grade and/or result of each test. With the advent of peer-to-peer (P2P), a disruptive technology that has its footprint in many Internet applications ranging from file sharing in Kazaa to a P2P telephony system with Skype, coupled with the advance in data mining techniques, more creative and helpful feedback models in e-learning can be established. By combining the power of the Internet and data mining technology, this paper proposes an educational feedback model in P2P based active e-learning that provides three feedback categories: intelligent feedback, statistical feedback, and interactive feedback.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank the people who have helped me to achieve this thesis.

Many thanks to Professor Xiao Su for having the patience to help, support, and guide me during the thesis period. Without her assistance, there is no way that I could have accomplished the thesis so successfully.

I would also like to thank Dr. Rod Fatoohi and Dr. Donald Hung for taking their valuable time to be the members of my thesis committee and to review my work.

In addition, I would like to thank my teammates, Anselm, Eric, Minyun, Pavithra, Ramya, Smita, Ty, and Yuqing, who worked on different aspects of the project for their valuable input during many long hours of meetings, discussion, and integration to make the entire project a complete solution.

Most of all, I must thank my family who supported me not only during the thesis period but also throughout the entire process of earning my master's degree.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Problem

The Internet is a large, complex information space that reflects much of the variety of our physical world. The advance in Internet technology and Internet applications has shaped our daily activities. In fact, more and more people utilize this technology daily to shop on-line, trade on-line, chat on-line, and even study on-line. Online education is very convenient to most people because they can obtain it easily without worrying about distance and time. Therefore, e-learning is becoming more and more popular and in high demand. However, the current e-learning technology based on the client-server model exposes several major problems that considerably hinder the power of e-learning.

First, the current e-learning system mimics a traditional classroom environment where the instructor is the server and students are passive recipient clients. In this model, instructors have to spend more time in coordinating with all the students. The many-to-one relationship inherent with the client-server model makes it difficult to scale. A small increase in the participating clients may require a massive upgrade of the server for more powerful resource handling and for bigger bandwidth. Besides, the centralized server is subject to a single point of failure in case the server goes down.

Secondly, the traditional client-server based e-learning model adopts the "push down" paradigm where information mostly flows from the server to the clients and seldom vice versa. This model also lacks a mechanism for clients to collaborate among themselves. Therefore, this model is inefficient in real time interaction between the students taking the class and the instructor, as well as among the students taking the class themselves.

Finally, the current e-learning model lacks an effective feedback system. Once students finish with a course, the usual feedback they obtain from such a model is the pass/fail record. Furthermore, seldom does the system reflect whether the tests given accurately assess the skill and knowledge sets required for the subject. The effective feedback system shouldn't restrict itself to assisting students in ways and methods to improve their overall learning process, but also should focus on the overall assessment of the tests given to determine if those tests correctly measure students' levels.

The peer-to-peer (P2P) based active e-learning model aims at providing the solution to all of the above problems utilizing peer-to-peer technology and data mining techniques.

1.2 Overview of P2P Based Active E-Learning System

The P2P based active e-learning system is one of the many breakthroughs in Internet applications. In general, the P2P based active e-learning system provides a virtual gigantic network infrastructure for effective academic activities where peers can communicate with each other, get information about any course offered, take tests, get certificates and feedback, and instruct their peers on topics in which they are certified.

The P2P based active e-learning system deploys the hybrid P2P network and consists of four main modules: the peer discovery the module, distributed content management module, peer collaboration module, and the peer certification and feedback module.

1.2.1 Peer Discovery Module

The peer discovery module, which resides on the directory server of the hybrid P2P network, facilitates resource sharing and initiates direct peer communication by hosting registered peer information, mainly peer IP addresses and peer shared index files. A registered peer, who wishes to communicate with other peers, first has to contact the directory server via a query. The peer discovery module constructs the response based on the query and its directory data. The initiating peer then directly establishes communication session(s) with other peer(s) based on the information from the reply.

The directory server in the P2P based active e-learning system is designed as a web service that utilizes XML based protocols to communicate with other systems. This approach has made servers independent of both operating systems and programming languages, and therefore facilitates the ease of adding peers to the network by the use of the open protocol. SOAP (Simple Object Access Protocol) is the primary communication protocol that connects peers to the directory server across the network.

1.2.2 Distributed Content Management Module

The distributed content management module manages course content, test materials and results, and the certification program. The MS server, which resides on the directory server or any other peers, stores and manages the data. This module is the core database management module of the entire e-learning system that tracks real-time course transactions, updates course information, keeps distributed contents in sync, maintains data integrity, and serves as input for the feedback daemon.

1.2.3 Peer Collaboration Module

The peer collaboration module provides mechanisms, such as text messaging and whiteboard tools, for peers to actively interact with one another. This feature makes the P2P based active e-learning system superior to the client-server based e-learning model since the client-server model lacks real-time interaction between participants that renders e-learning inefficient.

1.2.4 Peer Certification and Feedback Module

The P2P based active e-learning system develops a peer certification program whereby participating peers can take tests, get certified, and become certified instructors for that topic. In this model, peers can either be students, when they request/download course information or test programs from other peers; or instructors, when they are certified for the subjects and allowed to instruct; or both, when they are instructors for some classes and meanwhile students for some other classes.

The P2P based active e-learning system also provides both instructor peers and client peers with useful feedback. The feedback is statistically significant or intelligent predictions heavily based on data mining techniques. The feedback aims at improving the quality of the educational system.

1.2.5 Interaction among Different Modules

Following is the simplified view of the components and their interactions in a P2P based active e-learning system. Different interaction scenarios will be illustrated below.

The label attached to each picture in the diagram is used so that the interaction between different components can easily be illustrated. In fact, each picture in the diagram can be assigned a different label depending on the function it is currently participating in in the P2P network. The numbers in the diagram are used to illustrate the communication paths.

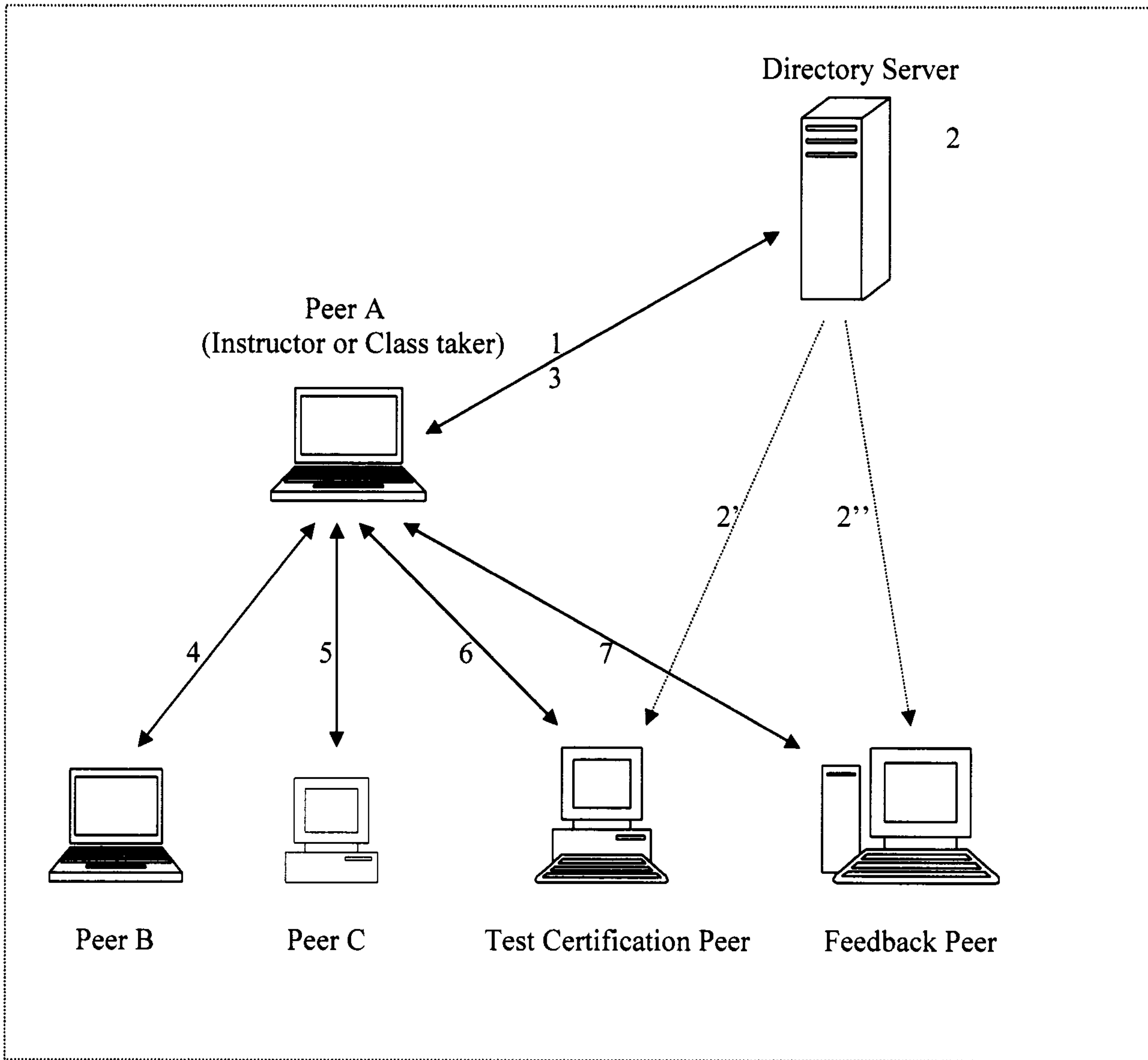


Figure 1: P2P Components and Interaction Diagram.

- Scenario 1: Peer A establishes a text messaging session with peer B.

Peer A first queries the directory server (1 in the diagram). The peer discovery module on the directory server constructs the response (2) containing information about peer B and replies to peer A (3). Peer A then establishes a direct text messaging session, which is managed by the peer collaboration module, with peer B (4).

- Scenario 2: Peer A establishes a whiteboard session with peer C.

Peer A first queries the directory server (1). The peer discovery module on the directory server constructs the response (2) containing information about peer C and replies to peer A (3). Peer A then establishes a direct whiteboard session, which is managed by the peer collaboration module, with peer C (5).

- Scenario 3: Peer A would like to take a particular test.

Peer A first queries the directory server (1). The query contains information about the particular test that A would like to take. Depending on whether the directory server or other peers in the network host this information, the peer discovery module can construct different responses. If the communication is assumed to be established with the directory server, the peer discovery module constructs a response (2) and replies to peer A (3). Peer A then connects to the directory server (3) to take a test. If the peer discovery module discovers a test certification peer that hosts this particular test, it will construct a response (2') and send the reply to peer A (3). Peer A then connects to the test certification peer (6) to take the test. The peer certification module takes care of generating the test and recording test results. The distributed content management module is responsible for keeping the updated data in sync across all nodes.

- Scenario 4: Peer A would like to get feedback.

Peer A first queries the directory server (1). The peer discovery module constructs the response based on the peer feedback information in its directory. Depending on whether the directory server or other peers in the network host this information, the peer discovery module can construct different responses. If the communication is assumed to be established with the directory server, the peer discovery module constructs a response (2) and replies to peer A (3). Peer A then connects to the directory server (3) to get feedback. If the peer discovery module discovers a feedback peer in the network, it will construct a response (2'') and sends the reply to peer A (3). Peer A then connects to the feedback peer (7) for getting feedback. The peer feedback module takes care of generating various feedback information based on the peer's input and the data managed by the distributed content management module.

In conclusion, the P2P based active e-learning system with effective applications, open architecture, and virtually "unlimited" resources sharing and interconnection based on the P2P paradigm will accelerate e-learning dramatically.

1.3 Scope

This paper focuses mainly on designing and implementing a fundamental feedback system of the P2P based active e-learning model, which is based on data mining technology. As a matter of fact, though data mining technology has been around for decades and its techniques have been widely and extensively adopted in profitable applications such as market research and business decision, few models are being applied in the educational field. This paper combines some of the very common methods and statistics used to propose a data mining technique, the k nearest neighbors with weighted selection algorithm, which is to be used mainly in its intelligent predictions.

Though the "K Nearest Neighbors with Weighted Selection" method can be applied for a vast variety of suggested feedback, this paper applies the proposed algorithm to provide feedback/update mainly on the tests given so that more accurate evaluation of the students' skills and knowledge can be assessed.

This paper also provides methods to graphically represent statistical significance so that users can easily make use of the data presented, as well as a mechanism for users to interact with one another via email in case questions/concerns arise during reviewing of the data. The user in this model is a peer who can be a student only, a peer instructor only, or both a student in some classes and a certified instructor for other courses.

1.4 Goal

The goal of this paper is to propose a data mining technique that can be implemented and evaluated through the empirical utilization of the model. The proposed technique can be further employed in building various academic predictions.

The goal of this paper is also to design and implement a practical feedback model in the educational system. The feedback system should not only provide instructive suggestions, but also effectively and intuitively represent data to users. The feedback system should provide feedback that can be classified into three main categories: the intelligent feedback, the statistical feedback and the interactive feedback. Figure 2 shows the overall picture of this feedback model. The model is not a complete feedback system but rather serves as a fundamental framework so that more useful feedback can be built and easily added based on these building blocks.

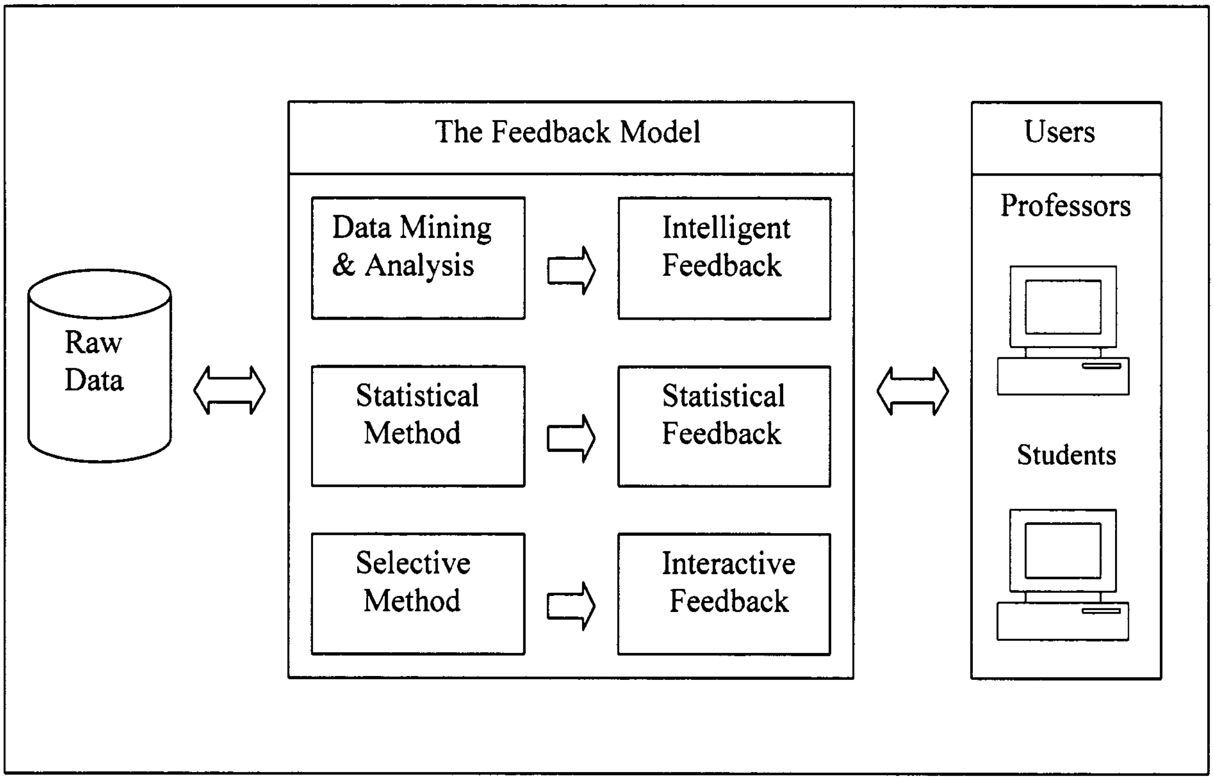


Figure 2: Feedback Model Overview.

CHAPTER 2
ARCHITECTURAL VIEW OF THE FEEDBACK MODEL

2.1 Three-layer Architecture

In order to facilitate working in the open network environment with various applications in the front end and database operation in the back end, the feedback model architecture is comprised of software components that can be classified into three main layers: the presentation layer, the core processing layer, and the data access layer. The multiple-layer architecture aims at providing modularity between different functional components for clarity in design and ease in maintenance. The architecture also provides a flexible developing framework and environment for future enhancements. The architectural diagram of the module is shown below.

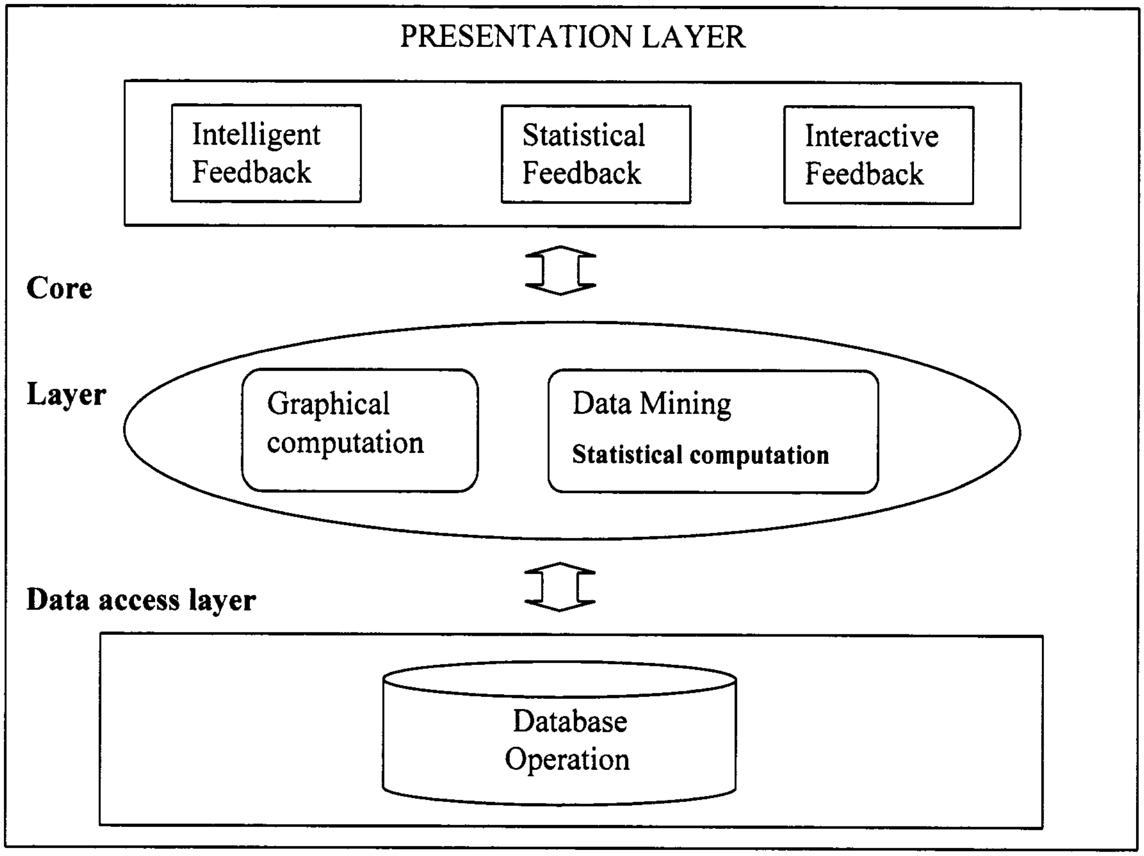


Figure 3: Three-layer Architecture of the Feedback Model.

2.1.1 The Presentation Layer

This layer is the front end of the feedback model where interaction between users and the system occurs. This layer has two main functions: to teach users how to operate the model and to effectively represent the result information. This layer therefore not only has to be user-friendly and self-conducting so that users just spend a minimum amount of effort to execute the system to obtain the maximum desired results, but also has to effectively present result information to users. In fact, no matter how sophisticated a model is, it will fail if the user interaction with the model is cumbersome and thus error-prone, or if users fail to interpret the result. Careful thought has been put into the design of the GUI such that it can smoothly guide users along and to provide users the ability to intuitively browse the model, review feedback, and input data if needed.

2.1.2 The Core-processing Layer

This layer is the core logical and computational unit of the system. The processing unit takes input from the presentation layer, and based on the type of the request, invokes the corresponding handlers to compute the result and then feeds this information back to the presentation layer. In many cases, the processing unit has to depend on the data provided by the data access layer below for computation. Thus this layer sits between the upper presentation layer and the lower data access layer and communicates to both of these layers to operate.

2.1.3 The Data Access Layer

This layer provides abstract operations to access and modify data without the burden of performing low level I/O operation. This layer provides appropriate data within its huge amount of data store to the core-processing layer for computation. It also receives instructions from the above layer to update the data if needed. This layer has imperative impact on system performance since the model is largely data driven. The choice of DBMS (Database Management System) and data structures plays a significant role in improving system operation.

2.2 Software Platform

To incorporate the three-layer architecture and the open environment, the feedback model is built on the Microsoft Application Platform. The Microsoft Application Platform provides the three main core technologies, the Visual Studio .NET, the .NET framework, and the Windows Server System, to facilitate building of the feedback model.

The Visual Studio .NET provides a high-level development environment for building applications on the .NET framework. The Visual Studio .NET facilitates editing, compiling, executing and debugging of applications in different programming languages. Therefore, though the core-processing layer in the feedback model is written in C#, more applications developed in a variety of programming languages can easily be integrated into the system.

The feedback model selects the .NET framework for its development environment because of the three main characteristics and features in the .NET framework. First, the .NET framework is a language-neutral component library and execution environment. The .NET framework not only provides a unified, object-oriented, extensible set of class libraries, but also creates a common set of application programming interfaces across all programming languages. This characteristic enables integration of cross-language applications into the current model. Second, the .NET framework provides Windows UI that is the basic framework for creating the GUI of the presentation layer. Windows UI with its rich set of tools has made the creation of the user interface of the model easy and graphically impressive. Finally, ADO .NET of the .NET framework, which is deployed by the data access layer in the feedback model to access the database, provides well-defined, factored components with predictable behavior, performance, and semantics that optimize database operation.

MS SQL 200 of the Windows Server is designed to fit into the .NET architecture and supports powerful data warehousing. The data access layer opens SQL query to the SQL server via ADO .NET and acts as a client to the SQL server. The SQL Server runs with several pools of threads. Some threads are dedicated to housekeeping tasks like logging, buffer-pool management, servicing operations requests, and monitoring the system. A second, larger pool of threads performs users' requests. These threads execute stored procedures or SQL statements requested by the clients.

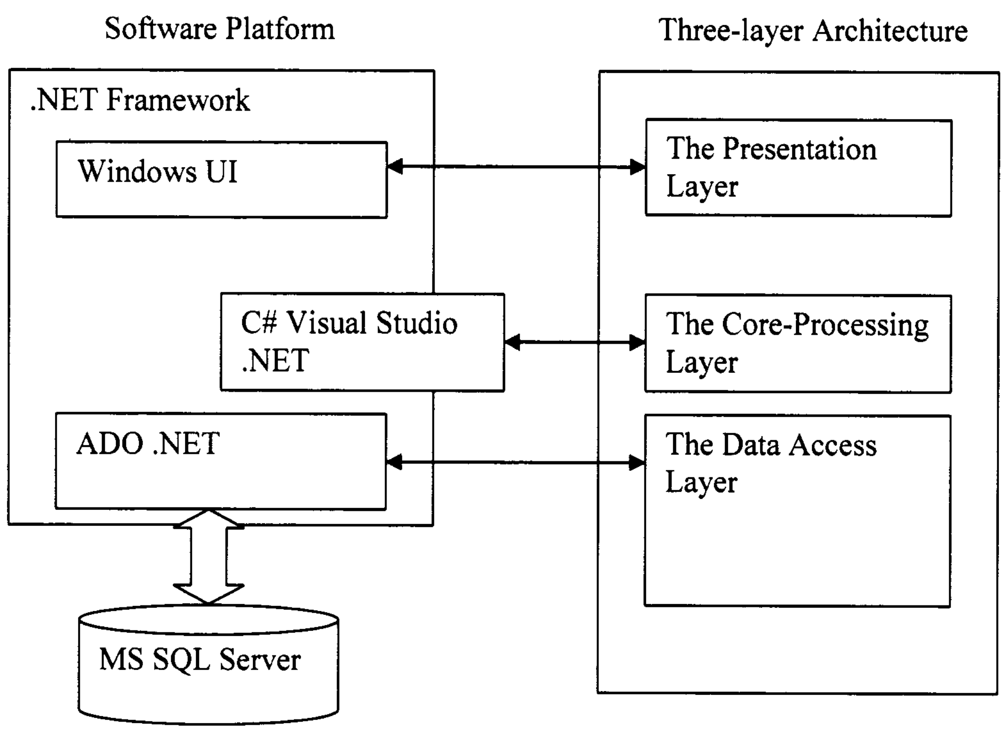


Figure 4: Relation Between the Three-layer Architecture and Platform.

CHAPTER 3
K NEAREST NEIGHBORS WITH WEIGHTED SELECTION

3.1 Overview

The feedback model in the P2P based active e-learning system relies heavily on data mining and data analysis to make predictions. Though nowadays data mining techniques have matured and been employed for many purposes, the choice of a data mining technique that best fits the application is critical. The reason is that though data mining is a powerful technology to retrieve hidden patterns and to formulate predictive information from the vast amount of raw data, the process can be very time consuming and costly.

To provide reasonable feedback for the system without complexity of implementation, the paper researches all the common data mining techniques and combines these techniques with statistics to build an algorithm, the k nearest neighbors with weighted selection, which is suitable for our scope.

3.2 Enabling Technologies

Our model is built based on the following technologies:

3.2.1 Massive Data Collection

With on-line education and the facilitation of the power of the Internet, distributive computing and file sharing, more data can be collected, stored, processed, and shared. The traditional data usually include such information:

- Student's personal record: age, sex, background, race, etc.

- School information
- Test results
- Student's major

With on-line study, some of these attributes can be added to the data store:

- Time spent studying each subject
- Pattern of study
- Material covered for the subject

3.2.2 Database Technology

Advanced database management systems have been developed to manage data efficiently. Relational databases (RDBMS) and Structured Query Language (SQL) are among the best choices in data access.

3.2.3 Data Mining Techniques

These techniques include the classical statistical methods such as histogram, linear regression, clustering, and k nearest neighbors as well as the common next generation techniques such as neural networks and rules induction. Below is a brief description of each technique.

- Histogram is a graphical representation of a frequency distribution in which the widths of contiguous vertical bars are proportional to the class widths of the variable and the heights of the bars are proportional to the class frequencies. The summary can quickly show important information about the database.

- Linear regression is the simplest form of regression which contains just one predictor and a prediction that linearly maps values from the predictors so that the lowest number of errors occurs in making a prediction.
- Clustering is a method that groups similar records together to give a high level view, or a bird's-eye view of the database.
- The K nearest neighbors method is the method whereby each interested record is classified based on a combination of the classes of the k record(s) most similar to it in the database.
- Neural network is a highly accurate predictive model that learns through training and resembles biological neural networks in structure, and thus can be applied across a large number of different types of problems.
- Rule induction is the establishment of if-then rules from the database mainly based on statistical significance.

3.3 Approach

There are three steps in building the proposed algorithm:

- Common statistical methods are applied to the database to extract emerging patterns, if any, so that the "weighted" value can be assigned to the parameter under consideration. In case there is no significant pattern emerging, other methods are used to determine the weighted value or a random (default) weighted value is assigned to the parameter under consideration.

- These assigned weighted parameters are sorted and stored in a reference table.
- The record under consideration is classified based on the k nearest neighbors method with the result set referencing the information from the reference table to make a decision.

3.4 Weight Calculation

Weighted values are determined using a combination of methods. Common methods such as histogram, clustering, and regression are applied to analyze data in order to detect patterns, if any, so that proper weighted values can be assigned to the attributes.

Weight calculation is the initial step and also the process that consumes the most time, effort and resources. In fact, many experiments and analyses of the data have to be performed in order to build any pattern out of the usually great amount of "unorganized" data.

3.4.1 Histogram

By counting the data based on the partition of the number of occurrences of one attribute in the database and graphically representing this finding, interesting information about the database may emerge. This information is the foundation for building weighted values. Below are two examples of counting the number of students based on how students score on a particular test. The first example shows the result of counting the number of male and female students that score above 90% and under 20% for the test.

The second example shows the result of counting the number of students with the same standard as the previous count but based on the percentage of the material covered for that subject. These graphs quickly show important information about the database such that the percentage of the material covered grows in proportion with the percentage of the score for the test while sex differentiation hardly makes any difference in scoring the test.

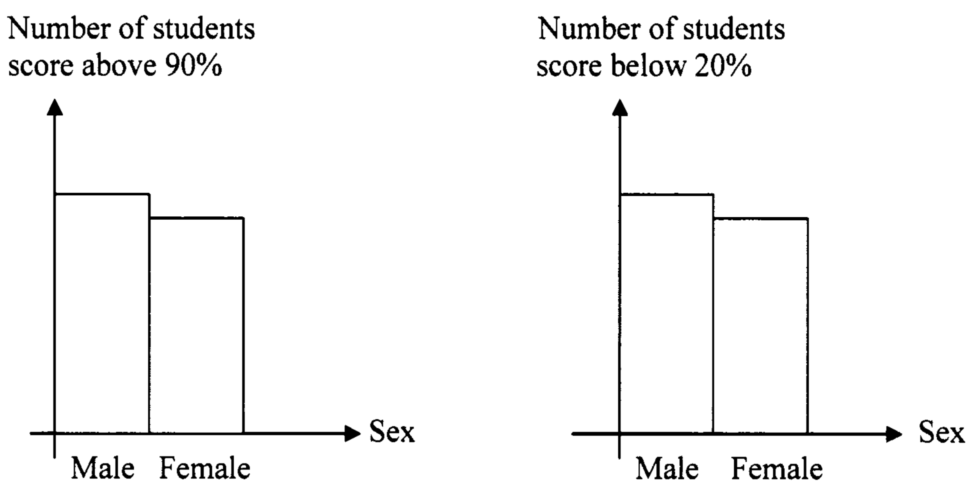


Figure 5: Student Sex in Prediction of Score.

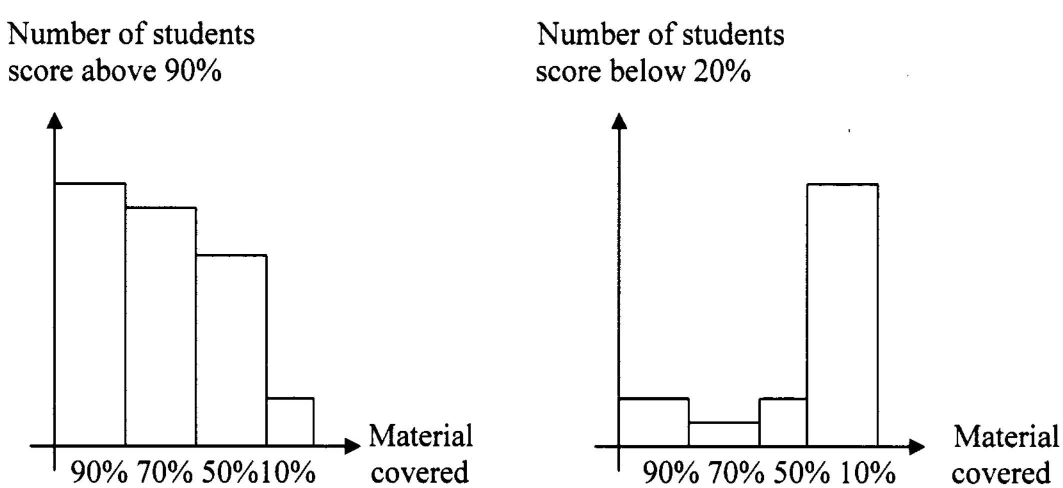


Figure 6: Material Covered in Prediction of Score.

3.4.2 Clustering

Data can also be analyzed by clustering. Clustering is the method of grouping similar records together to give a bird's-eye view of the database. Below shows a simple result of grouping students based on the percentage of their test scores for a particular subject. The result shows the same finding using a histogram.

Student ID	Age	Sex	Study Covered	Other Attributes
840	39	M	90%	...
849	24	F	50%	
4094	26	M	90%	
6879	25	M	90%	
7549	28	F	70%	
7638	30	M	80%	...
8010	27	F	80%	

Table 1: Clustering of Students Based on 90% Score for a Particular Test.

Student ID	Age	Sex	Study Covered	Other Attributes
110	29	F	10%	...
250	25	F	30%	
5100	36	M	20%	
6270	35	M	70%	
7301	38	F	50%	
8910	34	M	20%	...
9001	28	F	10%	

Table 2: Clustering of Students Based on 10% Score for a Particular Test.

3.4.3 Regression

Only linear regression is discussed. It is the simplest form of regression that contains just one predictor and a prediction. The relationship between these two can then be mapped on a two dimensional X-Y space with the predictor values plotted along the X-axis and records plotted for the prediction values along the Y-axis. In general, the relationship between these two parameters can be expressed in this linear equation: $y = a + b\,x$ where a and b are constants. In practice, a scatter diagram is plotted with the independent (predictor) values along the X-axis and the dependent (prediction) along the Y-axis. If these scatter points tend to cluster about the straight line, then there is a strong relationship between these two variables. In other words, the value of the independent variable can be used to estimate (predict) the value of the other.

3.5 Building Reference Table

This table contains the set of predictors under consideration categorizing based on weighted values. In case no weighted values are defined from the previous step, a default value (usually the mean value) is selected. Below is an example of the table containing the attributes (predictors) and their corresponding weighted values (from the results in step 1) that determine the result (measured by score) for studying a particular subject. This table, for example, lists "material covered" and "practice for the subject" as more influential factors in scoring the subject/test than age or ethnicity.

Attribute (Predictor)	Weight
Material covered	90
Practice	90
Time spent	80
Review	70
Related subject	60
Hand-on experience	50
Major	50
Age	1
Ethnic	1

Table 3: Example of the Reference Table.

3.6 K Nearest Neighbors with Weighted Selection

3.6.1 Algorithm

Prediction is based on k nearest neighbors to decide the order of importance (priority) determined by the weighted values. This simple algorithm works in the scope of this paper because:

- Most students experience the same learning process.
- Each subject (test) usually requires a set of knowledge and skills.
- A person has strong points and weak points that vary depending on the context (subject/test taken).

3.6.2 Building the Model

The algorithm can easily be implemented using the current common SQL in RDBMS technology. The model accepts user input, retrieves from this input special characteristics necessary for building the proximity standard, applies SQL rules, interprets the result set with reference to the reference table, and provides feedback to the user. Figure 7 below diagrams the architectural view of the model.

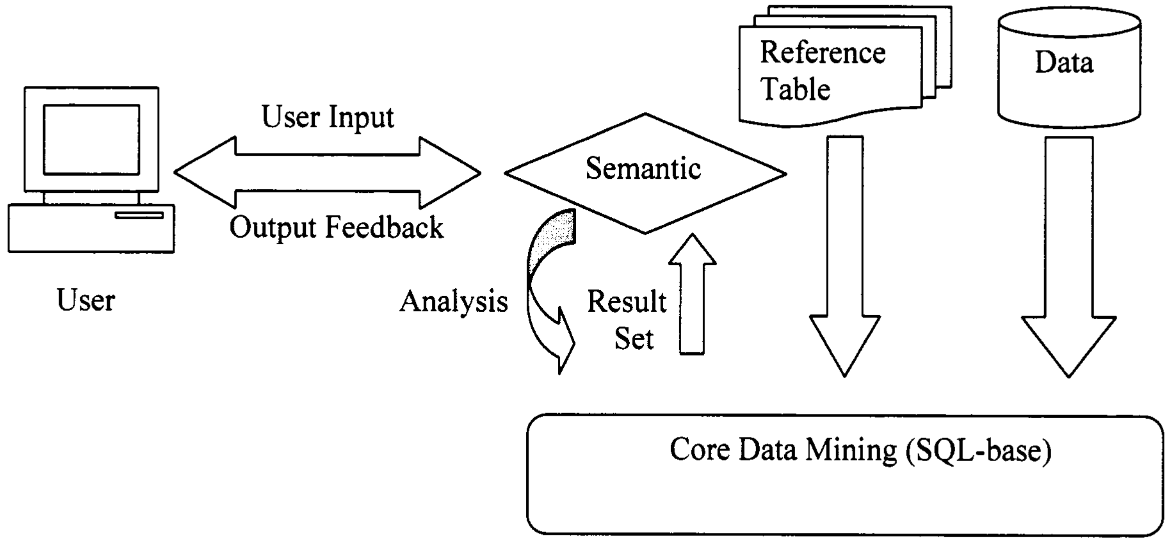


Figure 7: Architectural View of the Model.

In general, this process involves two major steps:

- Data retrieval:

  SELECT Attribute (*) FROM database WHERE conditions.

- Reference: The list of Attribute (*) from SELECT statement(s) referencing the Reference Table (priority list).

These two steps can further be refined into one simple query:

```
SELECT Attribute (*) FROM database, reference table
WHERE conditions
SORT by weights (priority)
```

3.6.3 Examples

Example 1: Use "K Nearest Neighbors with Weighted Selection" to predict how much effort is required to attain a certain grade in a subject.

Problem statement: A user would like to know in general how much effort is required in order to get a B in this subject.

Prediction formation process:

- The reference table of all the weighted attributes for this subject has been established during the data analysis phase.
- The model then selects the records with the score of B for this test. The result set contains a list of attributes with data about each attribute properly calculated (the mean, in this case).
- The suggestion (prediction) will list the characteristics (predictors) in order of the weighted values from the reference table.

Example 2: Use "K Nearest Neighbors with Weighted Selection" to update the difficulty level of an English vocabulary test.

Problem statement: An English instructor wants to know if the pre-assigned level for an English vocabulary word is accurate.

Prediction formation process:

- The reference table is built with each weight corresponding to a difficulty level assigned to each vocabulary word. For example, two tests below are actually devised by experienced educationists (600 Word Vocabulary Test).

TEST 1			
Level 1	Level 2	Level 3	Level 4
abroad	abandon	abridge	abhorrent
binoculars	ballot	aggregate	amorphous
daily	chaos	bivouac	crustacean
expedition	contraband	chronology	declivity
horizon	excavate	credulous	emaciated
jangle	fatigue	hireling	fabrication
limit	laboratory	indolent	galaxy
pattern	manual	meager	heretical
rate	purchase	nomadic	igneous

TEST 2			
Level 1	level 2	level 3	level 4
alter	beverage	biography	actuate
barometer	cardinal	decarbonize	bravura
distinct	demolish	domicile	comber
festival	graph	facet	gouache
hardship	humdrum	impunity	hieroglyphic
harpoon	impulsive	lore	hybrid
matinee	memorial	mercenary	iconoclast
reign	parallel	phantasm	maelstrom
report	terminate	restive	muezzin
waste	vivacious	taboo	resurgent

- The model then selects records with a difficulty level the same as that of the given word. The result set contains mostly statistical information about this level, such as number of correct/incorrect answers.
- Based on the statistical information, the difficulty level of the given word is then either justified or needs to be updated.

3.7 Confident Level

Given the nearest neighbor algorithm based on the following simple rule, "Objects that are 'near' to each other will have similar prediction values," the proposed model inherits these confident characteristics:

- The "distance" to the nearest neighbor provides a level of confidence. The closer the neighbor to the record under consideration, the higher the confidence of the prediction will be.
- The degree of similarity between the predictions within the k nearest neighbors can also provide a level of confidence. If most/all the nearest neighbors make the same prediction, the higher the confidence of the prediction will be.
- Statistical methods to assign "weighted" values provide another level of confidence to the prediction.

3.8 Model Validation and Refinement

- The model is tested with a pre-selected set of data. Should the model work, the observed result from applying this model should match the expectation from the selected data set.
- If any prediction is incorrect, it is re-evaluated by adjusting the weighted values as well as the condition of "proximity" of neighbors to this class. New standards and values may replace the existing ones once the whole process is re-applied with other predictions.

3.9 Extended Model

This paradigm can be applied to extend the model to provide a more comprehensive feedback model based on neural networks.

3.9.1 Characteristics

This extended model applies the neural network technique in data mining. Neural networks are similar to biological systems (or brains) that can detect patterns, make decisions, and learn. Thus the extended model inherits both advantages and disadvantages from the neural network model:

- The model is highly predictive for different applications.
- The model provides a thorough analysis of the system.
- The complexity of the model makes it difficult to use and deploy.

3.9.2 Building the Model

The neural network itself is so complex that the detailed implementation of the model is beyond the scope of this paper. Below only the important steps required to build the model are outlined.

- Determine the predictors (age, pre requisite, practice, material covered)
- Assign numeric values (weighted values) to predictors
- Translate the numeric output into the corresponding prediction.

3.9.3 Simple Example

Below is a simplified view of a neural network for predicting the score. The model accepts the values for the predictors on the input nodes. These valued nodes are then multiplied by values that are stored in the links. These values are then added together at the output node at the far right. The resulting number is the prediction based on a special standard, such as 0 to be unacceptable and 1 to be good score. The considered parameters in the example with their corresponding weighted values are shown in Figure 8.

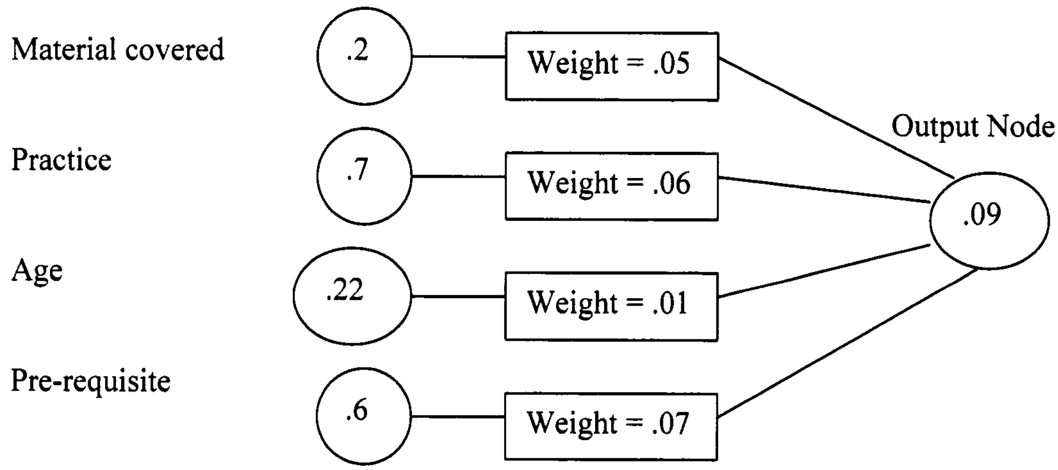


Figure 8: Simplified View of the Neural Network Model with Four Predictors.

The output in this figure is 0.09, nearer to 0, which means the person with the corresponding values for these attributes (age, practice, material covered, pre-requisite) is likely to get a low score for this particular subject.

3.10 Conclusion

By combining common data mining techniques and statistics, the "k nearest neighbors with weighted selection" is simple to implement but provides enough predictions (feedback) in the specified scope. Furthermore, by adopting these techniques to build a complete neural network for the educational model, a thorough and intelligent analysis of the data can be achieved so that more generic prediction/feedback can be provided to users while the complexity of building the model can be alleviated due to the weighted values that are assigned for each link and node. Weighted values based on statistical calculation reduce the empirical process in assigning the correct values to the links and nodes of a workable neural network model.

CHAPTER 4
THE FEEDBACK MODEL

The core-processing layer implemented in C# provides a logical and computational center for the three main functional components that structure the feedback model: the intelligent feedback, the statistical feedback, and the interactive feedback.

4.1 Intelligent Feedback

4.1.1 Overview

The proposed intelligent feedback in this model will utilize the k nearest neighbors with the weighted selection method to retrieve useful predictions and intelligently provide practical feedback to users.

To best evaluate test takers' skills and knowledge of the topic, the topic consists of different tests. Each test includes different questions, and each question is assigned a difficulty level. If the test takers pass more questions with higher difficulty levels, their levels more likely match the level that topic requires. Therefore questions with a higher difficulty level have more weight than those with a lower difficulty level in determining test takers' skills for that topic. This difficulty level is the weighted value in the algorithm, and the test takers who pass the same level are "similar" in their skill sets for the test.

4.1.2 Feedback Scope

The intelligent feedback focuses on intelligent predictions to provide feedback/update mainly on the difficulty level of the tests given so that a more accurate evaluation of the students' skills and knowledge can be assessed.

4.1.3 Assumptions

The algorithm assumes the following conditions and presumptions:

- Sample data is large enough for confidence level.
- Question levels with a higher index are more difficult than those with a lower index for the same test.
- The same level questions should have the same difficulty level.
- Test takers with a similar level should have almost the same correct rate (k nearest neighbors).

4.1.4 Algorithm

In general, the algorithm calculates the mean of the percent correct rate for all the questions with the same level as the difficulty level of the question under consideration. The calculated result is then used to compare with the percent of the correct rate of this question to determine if the difficulty level assigned to this question is accurate.

The algorithm consists of the following steps and computations:

- Weighted values are the difficulty levels already assigned to each question.
- The number of correct answers from test takers and the total number of test takers who took this question are selected to calculate the percentage of the correct rate of the question under consideration.

- The number of correct answers, and total number of test takers who took other questions with the same level as the question under consideration are selected to calculate the percentage of the correct rate of other questions of the same level.
- The mean of the percentage of the correct rate of questions one level above and one level beneath the level under consideration are also calculated.
- These calculated numbers are then compared with each other to determine the correct level of the question under consideration.

4.1.5 Validation

This algorithm can be validated by:

- Empirical approach: As more users participate in the system, more data are being collected and tested against the new difficulty level. Therefore, the algorithm is self-corrective and can be validated via, for instance, analyzing the statistical data.
- Controlled approach: Selectively chosen candidates who have high enough qualifications participate in the test after the new difficulty level is set; the data collected in this case are more controlled and can be used to validate the algorithm.

4.2 Statistical Feedback

Raw data in the system database are selectively calculated and presented to users both statistically and graphically so that a more general view of the data can be achieved as well as the fact that more useful information or conclusions can be deduced.

- The mean and standard deviation of student grades for each test are computed.

- The detailed data about the test are also presented.

In addition, the statistical data can also be used to validate the algorithm employed in the intelligent feedback to determine the difficulty level of a question. Ideally if a large enough selected sample of test takers is chosen to take the test after the new difficulty level is set, the data collected are more controlled and can be used to validate the algorithm.

4.3 Interactive Feedback

Interaction is an inherent element in any educational system. In other words, an educational system cannot be effective if there is only one flow in direction of information. The feedback model in the P2P e-learning system provides two types of interactive feedback: the passive (or machine) and the active (or human) feedback.

- The passive (or machine) feedback: This interactive feedback is so-called because the feedback is based on organizing the raw data in the system and selectively presents the information to the user. For instance, test takers would like to review details about a particular test taken such as, reviewing the questions, giving answers to them, and learning the correct answers.
- The active (or human) feedback: This term is a little bit misleading since the system itself doesn't provide "human" feedback but only provides a handy mechanism for the test taker to get information on what is bothering him/her. For instance, the user while reviewing these questions may pop up some further questions, concerns, or ideas for interested peer users. This user can just simply record the question or concern in an email and send this mail to any interested

participants. SmtpMail class, which belongs to System.Web.Mail namespace of the .NET framework, is used to deliver email messages through the SMTP mail service built into Microsoft Windows 2000 or through an arbitrary SMTP server using the Collaboration Data Objects of Windows 2000 (CDOSYS) message component. A code snapshot of this class is shown below:

```
using System.Web.Mail;

// basic authentication
mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/s
               mtpauthenticate", "1");
// set your username here
mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/s
                endusername", "<username>");
// set your password here
mail.Fields.Add("http://schemas.microsoft.com/cdo/configuration/s
                endpassword", "<password>");
// your mail server goes here
SmtpMail.SmtpServer = "smtpauth.mailserver.net";
// send mail
SmtpMail.Send(mail);
```

CHAPTER 5
DATABASE DESIGN

The data access layer is the abstract layer for direct I/O access to the database. Because of the frequency of data access and the amount of data to be processed, this layer has to be carefully structured to avoid bottlenecks caused by I/O operation. Therefore, decisions on data structure and DBMS to meet this requirement are carefully made. The data access layer utilizes ADO .NET technology to communicate with MS SQL server.

5.1 Relational Database

The data layer of the feedback model adopts the relational approach in structuring the data. There are three main advantages for this approach. First, the MS SQL server and ADO .NET technology support relational databases and the common SQL. Secondly, relational databases satisfy an important property: "The entire information content of the database is represented in one and only one way, namely as explicit data values" (Date 55). Finally, the constraints among data and data tables make data atomic and thus easy to control and manage.

The feedback model has all data grouped into tables. Though the entire P2P active based e-learning system has more data, only data of interest for the feedback model are discussed. Below is a brief description of each table.

- The *Test* table records basic attributes about that particular test, such as test time, test due date, etc.

- The *QuestionAnswer* table contains questions, answer choices, correct answers, and some other useful information about the tests such as the number of correct answer and the number of attempts.
- The *MemberAnswer* table contains information that is related to a student for a specified test such as student ID, test ID, question ID, etc.
- The *MemberTest* table contains details about a student for a specified test, his/her correct/incorrect answers, grade, etc.
- The *Certification* table contains information about the certified title for a particular test, such as information about user ID, passing grade, on date issued, etc.

5.2 Database Diagram

The feedback model relies heavily on the database to provide predictions or useful feedback, therefore the data model in the feedback module focuses not only on what data should be stored in the database but also on how data can be efficiently processed. To meet this objective, the feedback model applies the relational approach in grouping related attributes into tables and identifying the relation between attributes in different tables via the use of primary and foreign keys. These keys are the most basic components of relational theory. The primary key (PK) imposes entity integrity by uniquely identifying entity instances. The foreign key (FK) enforces referential integrity by establishing a connection between two entities.

Figure 9 diagrams the tables with their associated attributes and corresponding primary key(s) and foreign key(s), if any, assigned to the attributes. The arrows in the diagram show the association established by these keys between different tables.

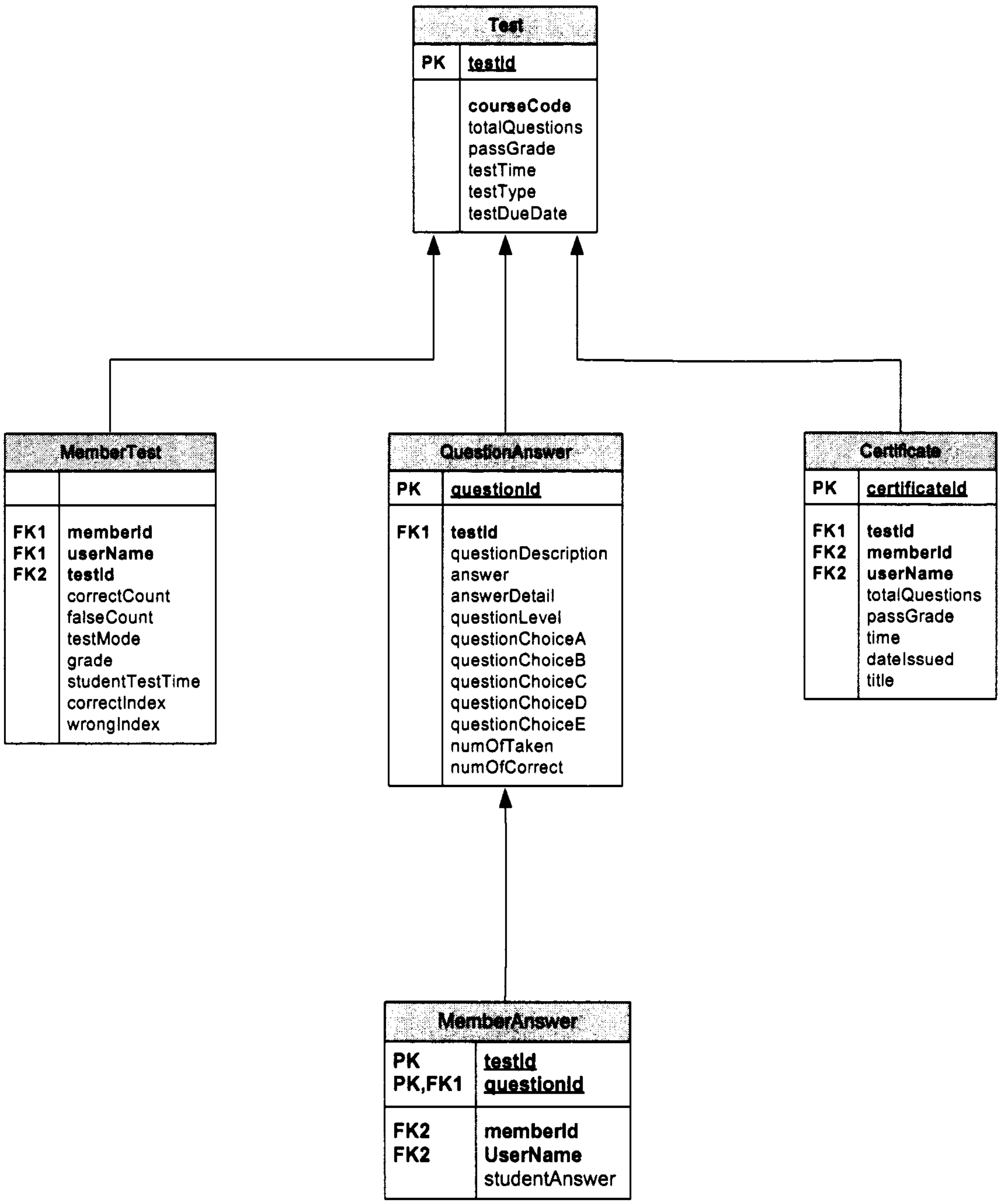


Figure 9: Database Diagram.

5.3 Database Operation Using ADO .NET

5.3.1 Database Connection

This step is required to initiate a connection to the database prior to performing any database transaction. An instance of OleDbConnection is used to maintain a unique connection to the data source. Besides proving thread safe for public static member instances, OleDbConnection is selected because 1) the .NET framework Data Provider for OLD DB automatically manages connection pooling and thus eliminates high-performance applications from the task of self-managed connection pooling, and 2) security properties can be assigned to the connection so that any caller to the application with an OleDbConnection instance is required to have adequate permission to be successful. Below is a code snapshot using the OleDbConnection method.

```
public void databaseConnection()
{
  //------Start to connect the database ----------------------------
  OleDbConnection conn;
  string sqlStmt;
  conn = new OleDbConnection(@"Integrated Security=SSPI;
             PacketSize=4096;
             Data Source=""(local)"";
             Tag with column collation when possible=False;
             Initial Catalog=P2P;Use Procedure for Prepare=1;
             Auto Translate=True;Persist Security Info=False;
             Provider=""SQLOLEDB.1"";Workstation ID=CAROL;
             Use Encryption for Data=False");
  // For cmbTestId
  sqlStmt = "SELECT DISTINCT testId FROM memberTest where userName = '"
             + name + "'";
  cmbSelect(conn, cmbTestId, "memberTest", "testId", sqlStmt,
            "testID");
  cmbTestId.SelectedValueChanged += new
                  EventHandler(cmbTestId_SelectedValueChanged);
  cmbTestId_Function(conn);
  conn.Close();
} // end of databaseConnection()
```

5.3.2 Database Transaction

Once a connection to the data source is established, an instance of OleDbDataAdapter is created and will serve as a link between the DataSet and the data source for retrieving and saving data. DataSet is an in-memory cache of data retrieved from the data source and is a major component of the ADO .NET architecture. DataSet improves performance dramatically, especially when a large volume of data is transferred. OleDbDataAdpater provides all the necessary mechanisms to keep the data between the links in sync. The "Fill" method is used to load data from the data source into the DataSet, and the "Update" method is used to update changes made in the DataSet back to the data source. Below is a code snapshot using OleDbDataAdapter.

```
private void cmbSelect(OleDbConnection conn,
                 System.Windows.Forms.ComboBox cmb,
                 string tableName, string attb,
                 string sqlStmt, string field)
{
      OleDbDataAdapter da;
      DataSet ds = new DataSet();
      da = new OleDbDataAdapter(sqlStmt,conn);
      da.Fill(ds, tableName);
      cmb.DataSource = ds;
      cmb.DisplayMember = tableName + "." + attb;
      cmb.ValueMember = tableName + "." + attb;
      cmb.DataSource = utilMyCmbBox.order(ds, field, cmb.Items.Count);
      cmb.DisplayMember = tableName + "." + attb;
      cmb.ValueMember = tableName + "." + attb;
      da.Dispose();
} // end of cmbSelect()
```

CHAPTER 6
GUI DESIGN

The principal component of the presentation layer of the feedback model is the GUI. Because the presentation layer is the front-end where users interact with the system, it is important for this interface to be user-friendly, intuitive, and artistic. The presentation layer is built on a Window UI within the .NET framework, which provides sufficient libraries to make this happen.

6.1 Panel

The two main frameworks of the feedback system, the instructor feedback screen and the student feedback screen, are built on the windows panel control that belongs to the System.Windows.Forms namespace. The panel can contain other controls so that related features can be grouped together inside the panel to make up a complete entity.

6.1.1 Instructor Feedback Panel

The instructor feedback panel contains features and controls related to the feedback for instructors. Below is the screenshot of the instructor feedback panel.

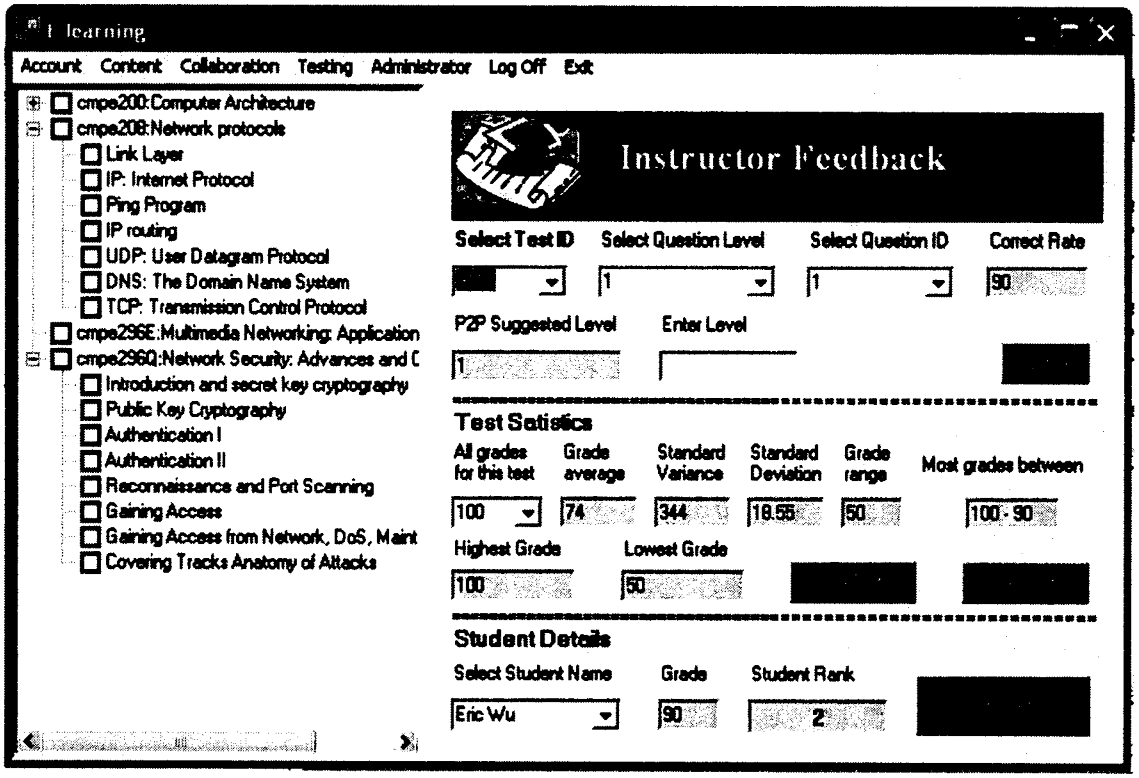
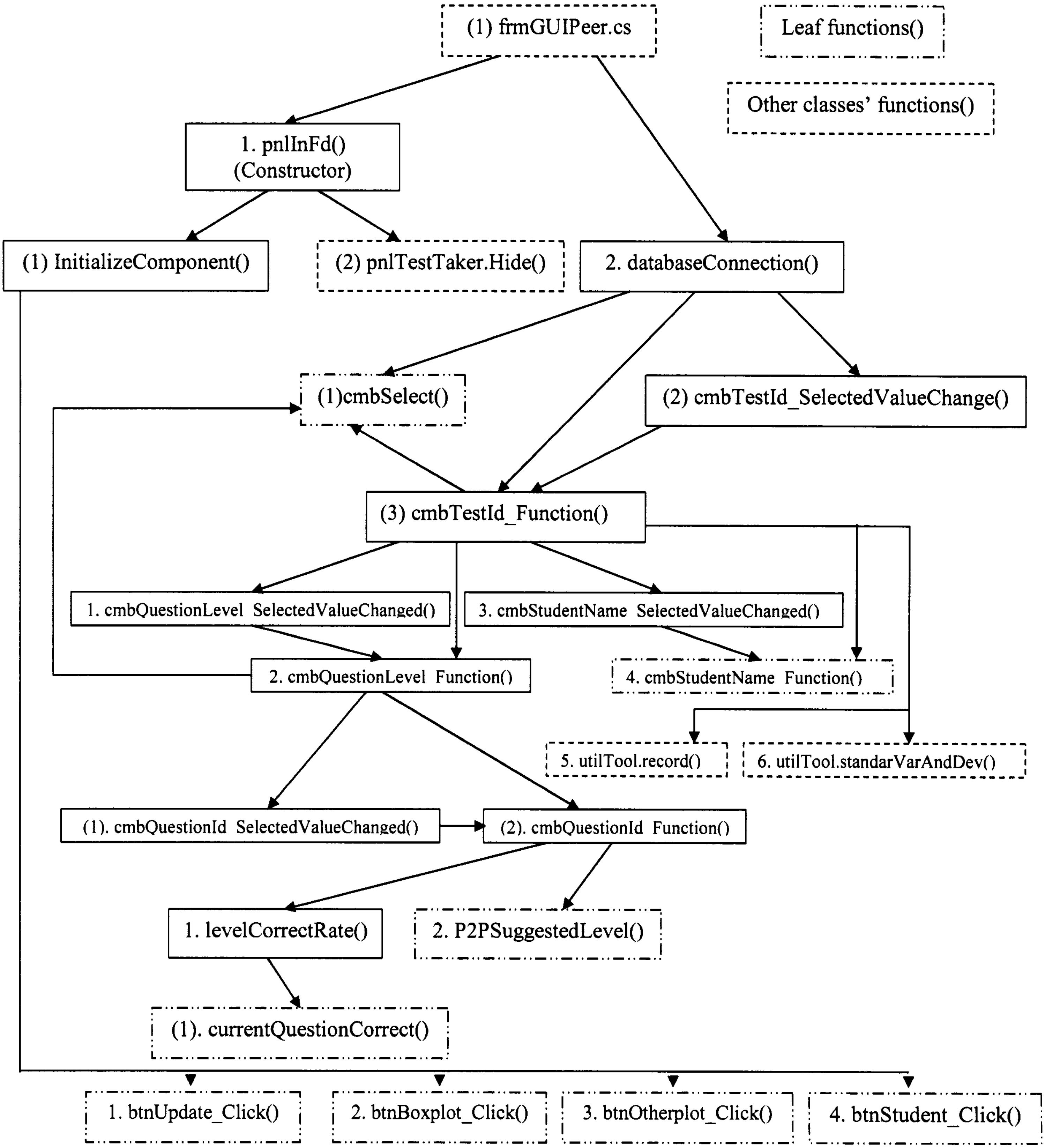


Figure 10: Instructor Feedback Panel.

Following is the control flow diagram of the instructor feedback panel. The main menu form (frmGuiPeer.cs) initiates the instructor feedback panel by creating an instance of pnlInFd panel in the front end and connecting in the back end to the database by calling the databaseConnection method. The pnlInFd panel then initializes all the components, which include labels, text boxes, combo boxes, buttons, and the picture box. All these entities are grouped under the parent instructor feedback panel to provide controls and feedback for instructors. The picture box displays the "Instructor Feedback" banner. The various labels are used to explain the usage of different text boxes and combo boxes. While text boxes are used to display only one selected item, combo boxes are list boxes that can store many items at once with the current selected item displayed in the list box portion of the combo box. Buttons are controls that can entail an action. The four buttons in the panel are the "update" button to update the difficulty level of a question, the "box plot" button to draw graphs, the "other plots" to draw other pictures, and the "student detail" button to invoke the student panel button. The database connection branch represents functions of the core-processing layer and the data access layer. This branch connects to the database, calculates the difficulty level of the selected question ID, and provides feedback to the presentation layer. The presentation layer will then feed the data into the various interface entities in the main instructor feedback panel. The branch is included in the diagram for completeness of the picture.

Figure 11: Instructor Feedback Control Flow.



6.1.2 Student Feedback Panel

Similarly, the student feedback panel contains features and controls related to the student feedback. Below is the screenshot of the student feedback panel.

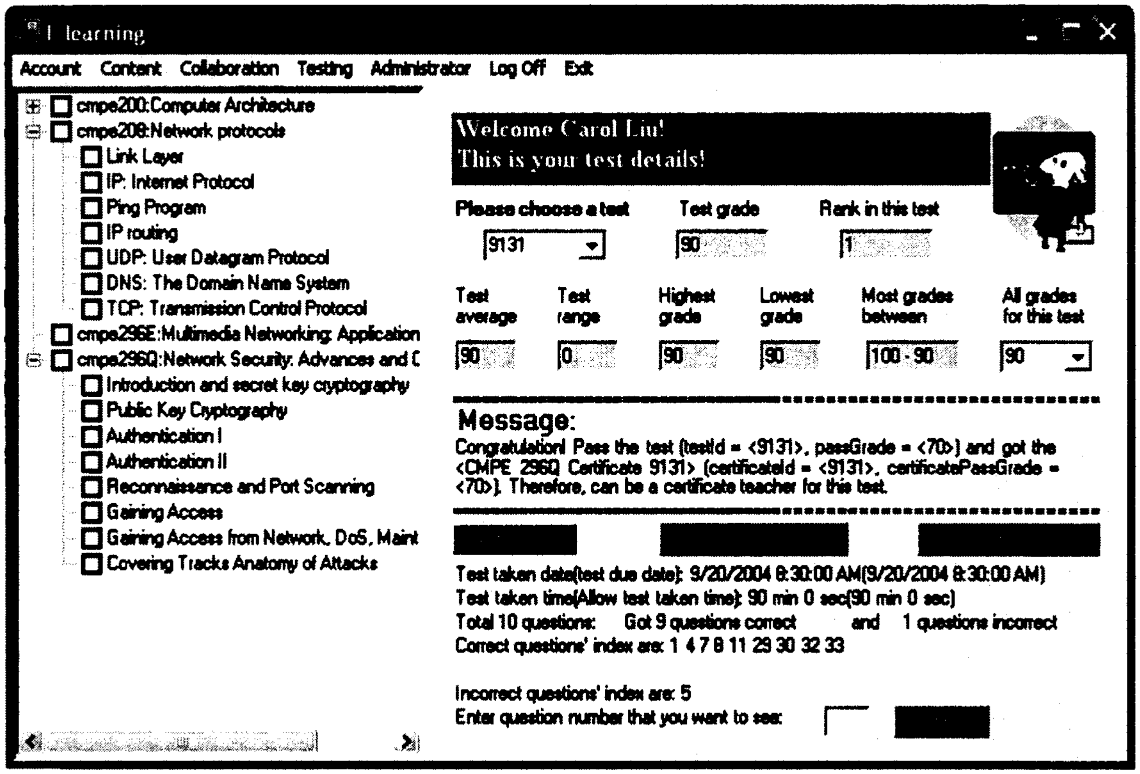
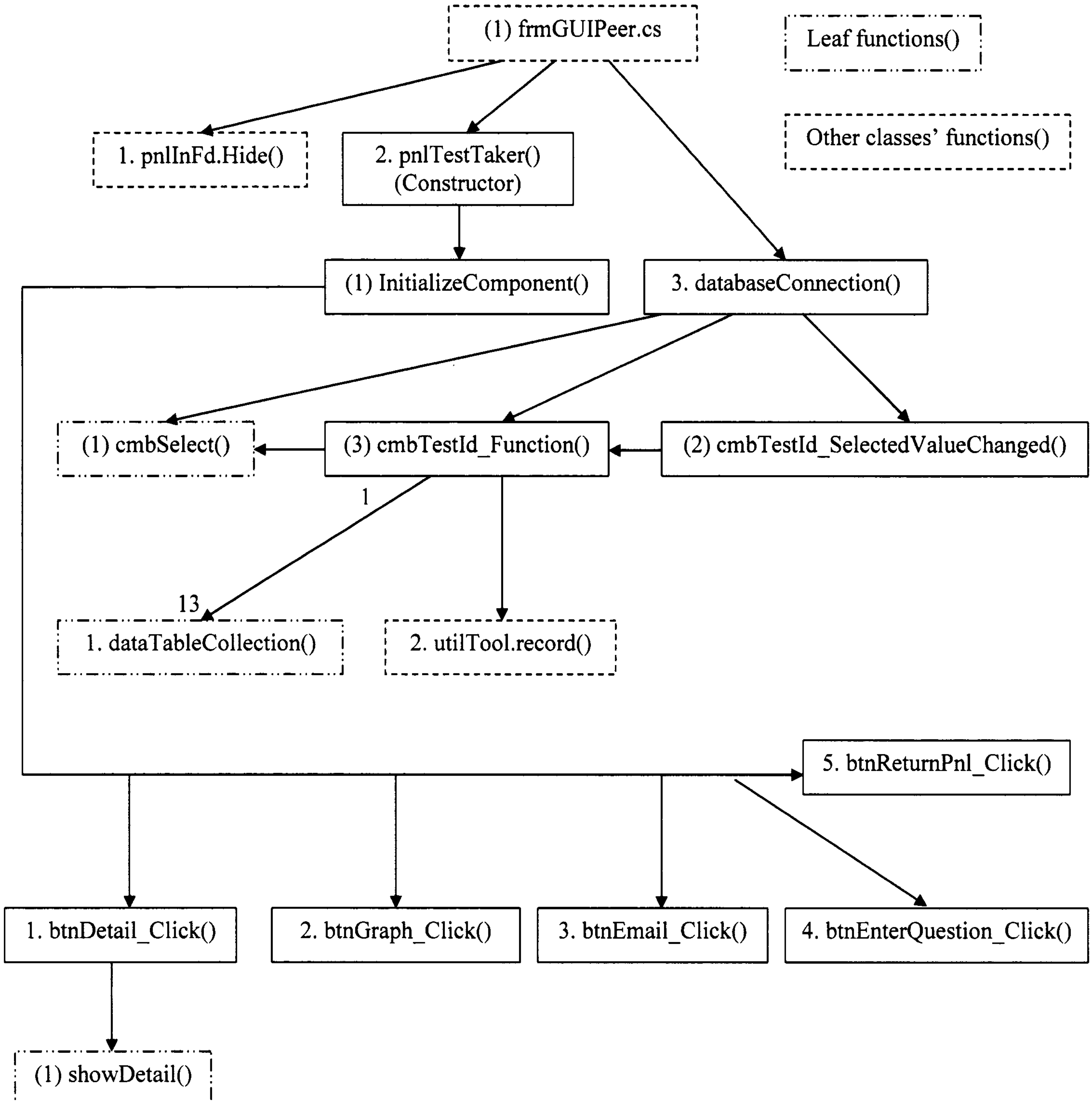


Figure 12: Test Taker Feedback Panel.

Following is the control flow diagram of the student feedback panel. This diagram is very much similar to the instructor feedback diagram. The main menu form (frmGuiPeer.cs) initiates the student feedback panel by creating an instance of the pnlTestTaker panel. The pnlTestTaker panel then initializes all components of interest.

Figure 13: Test Taker Feedback Control Flow.



6.2 Form

Various dialogs in the feedback model use the window form that also belongs to the System.Windows.Forms namespace. The window form can also contain controls and can be set for a specified size to display at a particular position.

6.2.1 Question Dialog

The question dialog contains information about the description of the question and the test taker's choice for the question. Below is the screenshot of the dialog.

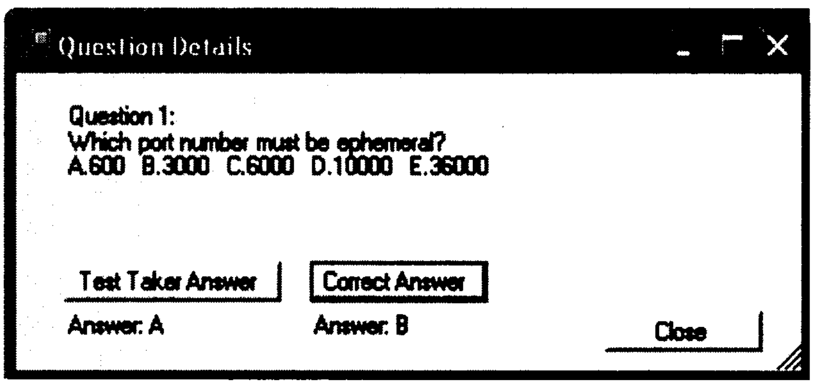
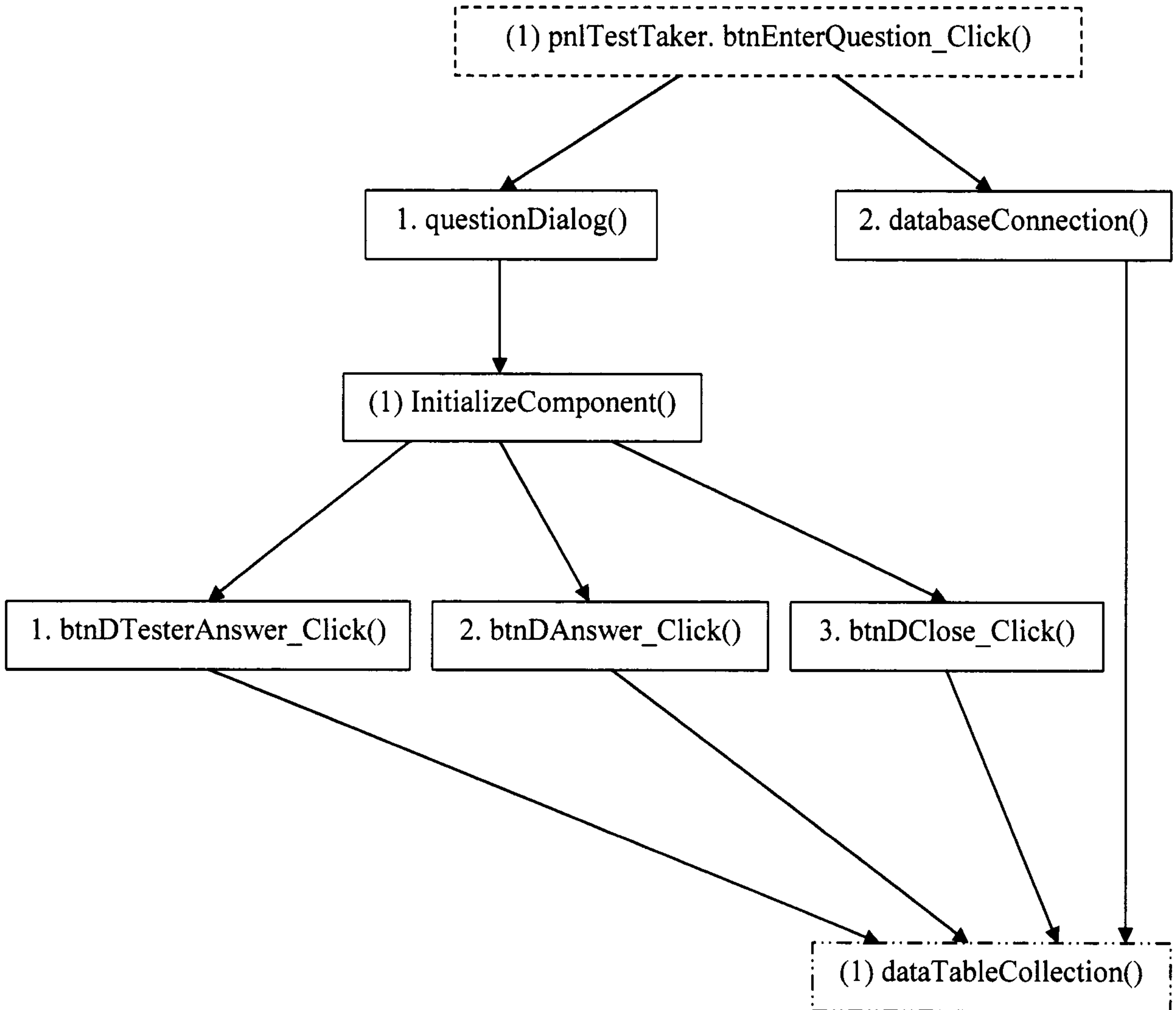


Figure 14: Question Dialog.

Following is the control flow diagram of the question dialog. The question dialog is instantiated by the "submit" button in the student feedback panel. This dialog then initializes all related components detailing in Figure 14.

Figure 15: Question Dialog Control Flow.



6.2.2 Email Form

The email form displays the dialog for sending the email message and is shown below.

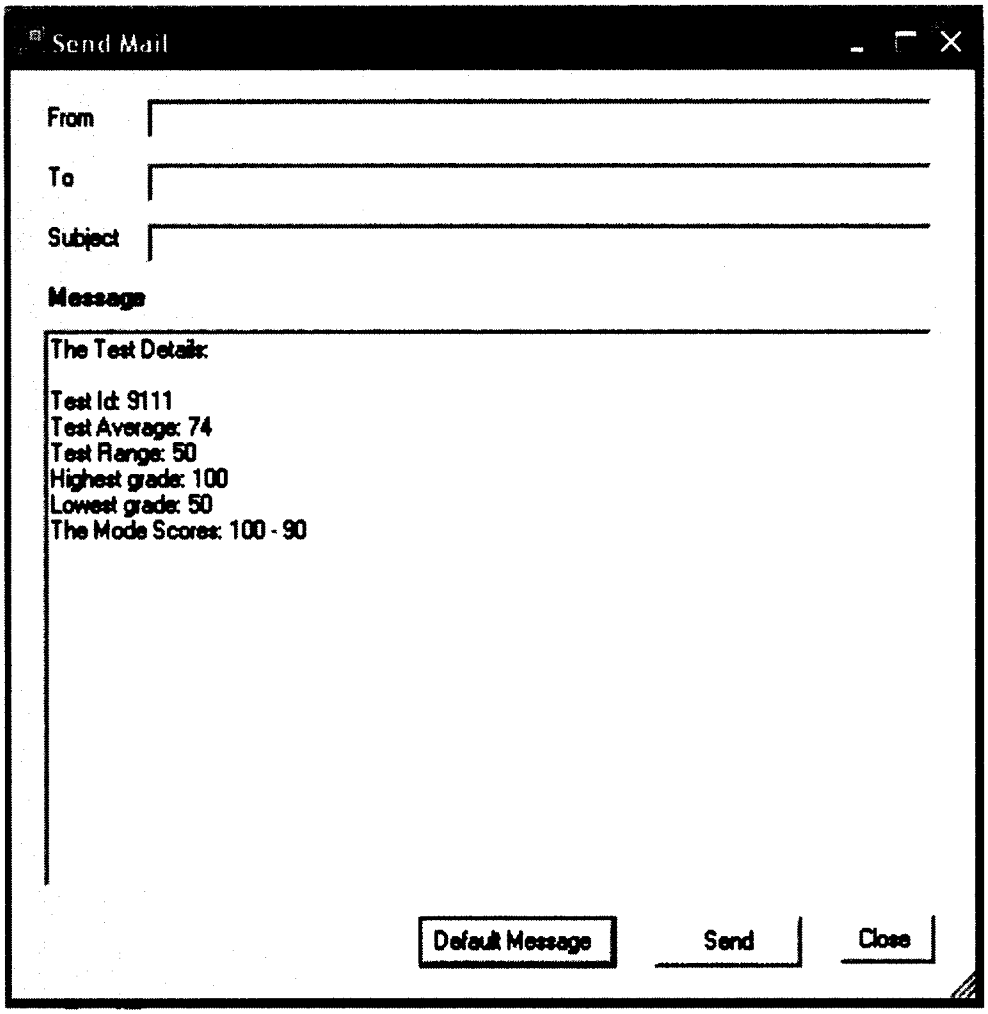
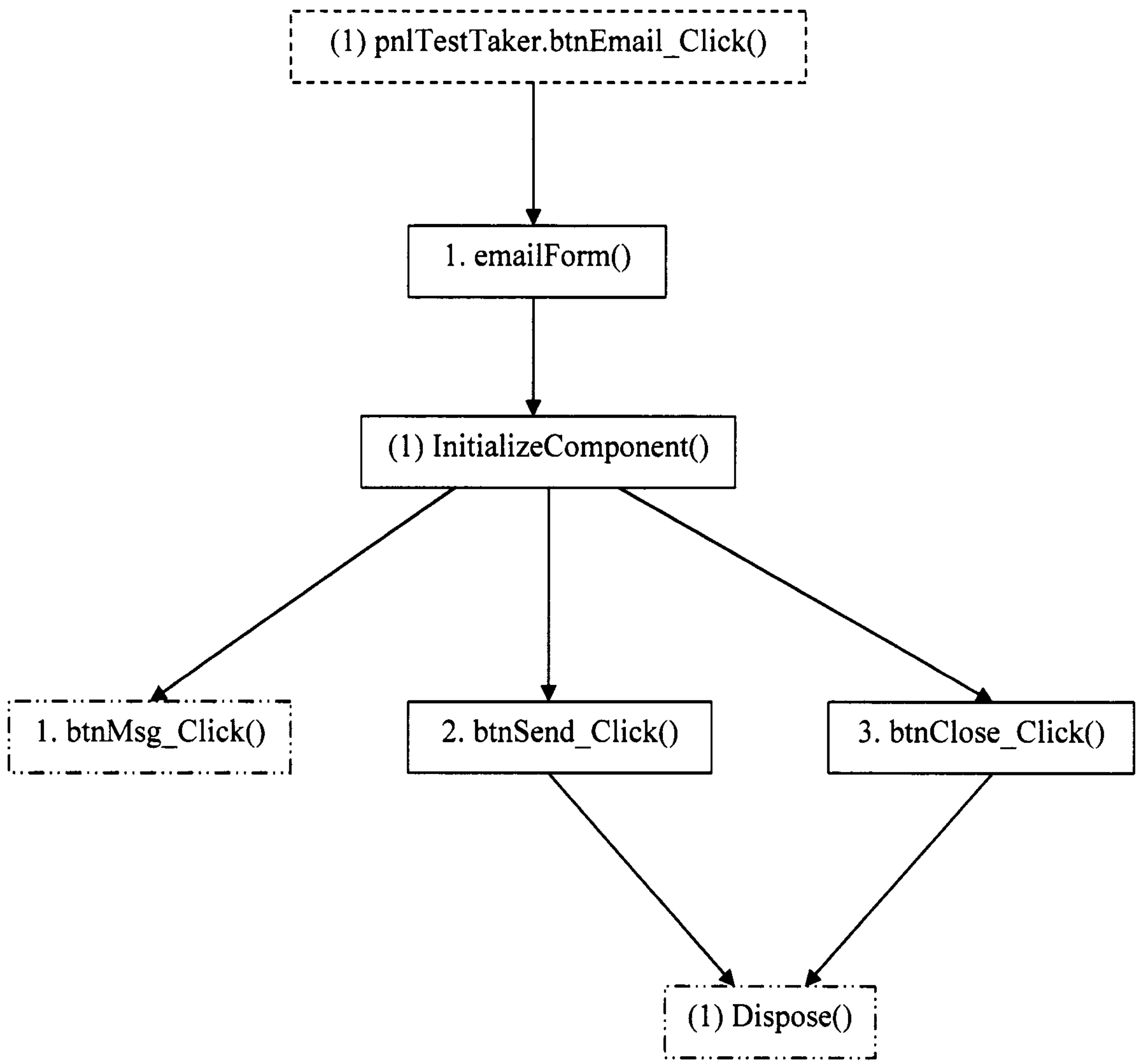


Figure 16: Email Form.

Following is the control flow diagram of the email form. The “Send Email” button invokes the email form. This form then initializes all related components.

Figure 17: Email Control Flow.



6.3 Graph

The feedback model utilizes graphic libraries and dialogs to display many informative graphs. Graphs are techniques used by the presentation layer to provide users with an instant summary and general picture of the data.

6.3.1 Graph Dialog

Graph dialog is a form instance used to display graphical objects. The screenshots of different graphical objects are presented and followed by the control flow diagram.

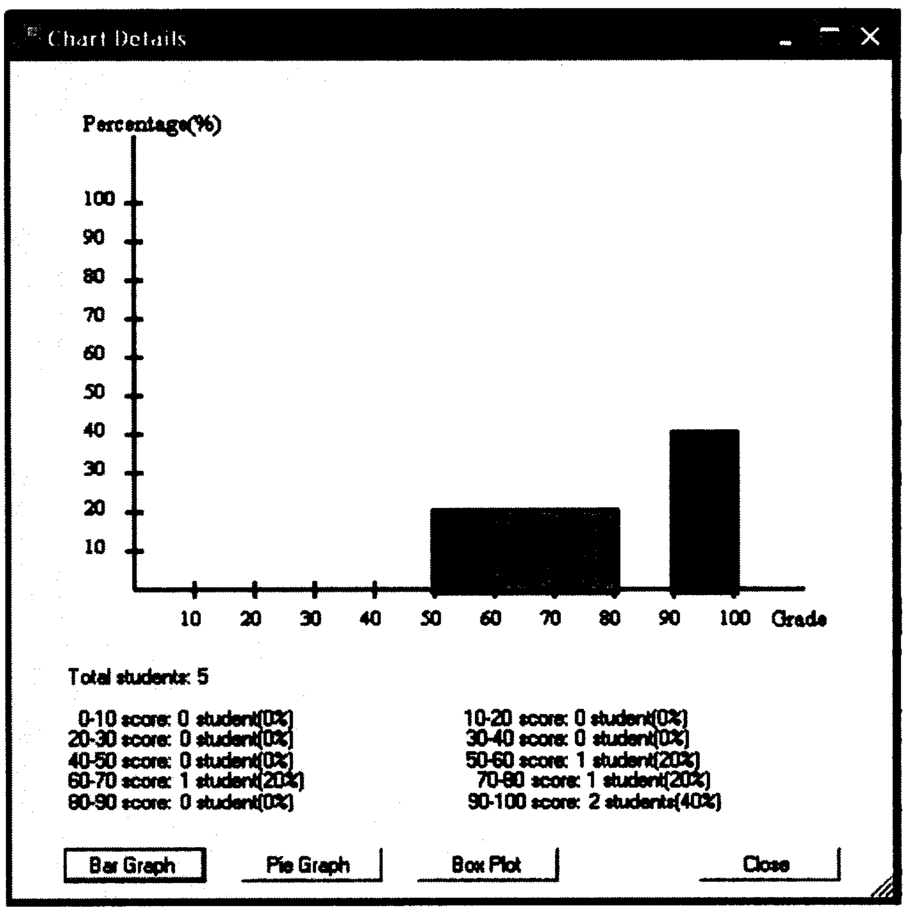


Figure 18: Bar Graph.

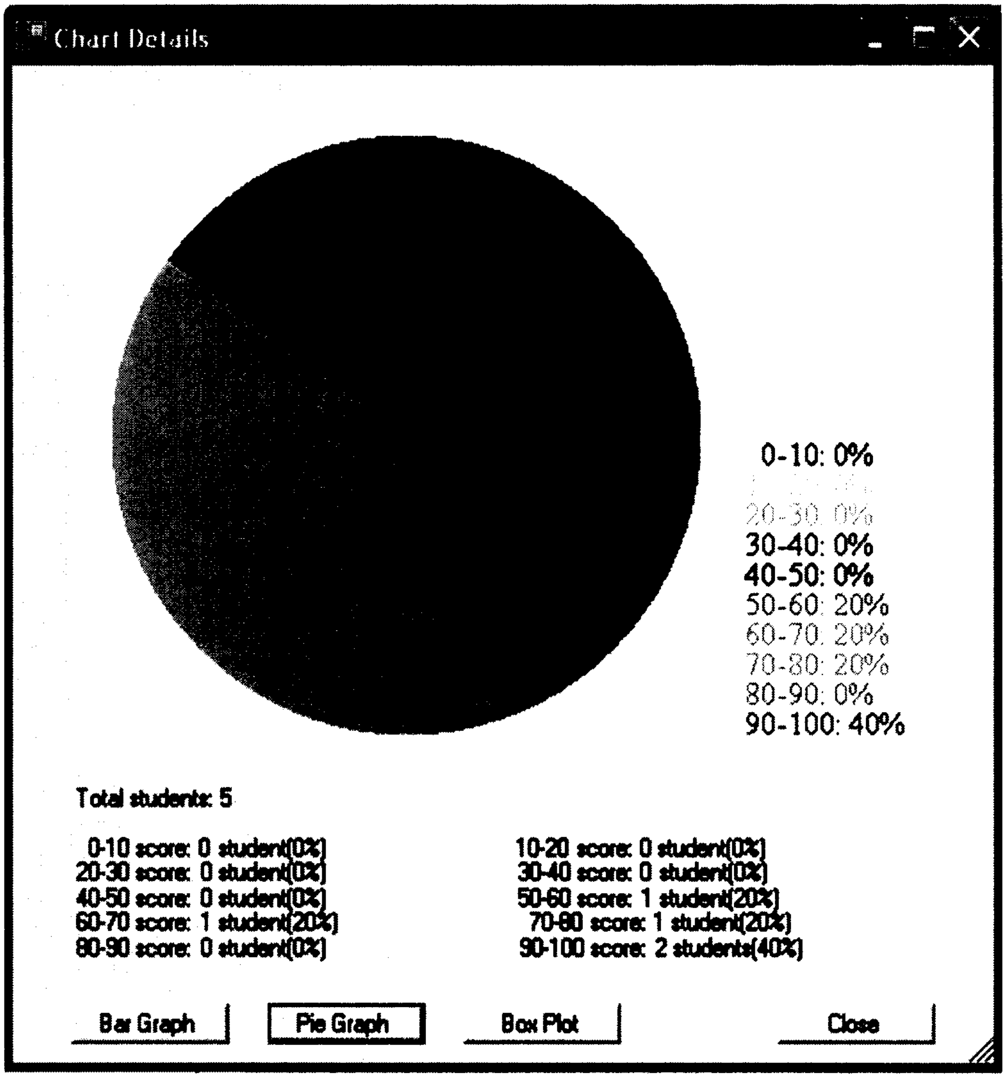


Figure 19: Pie Graph.

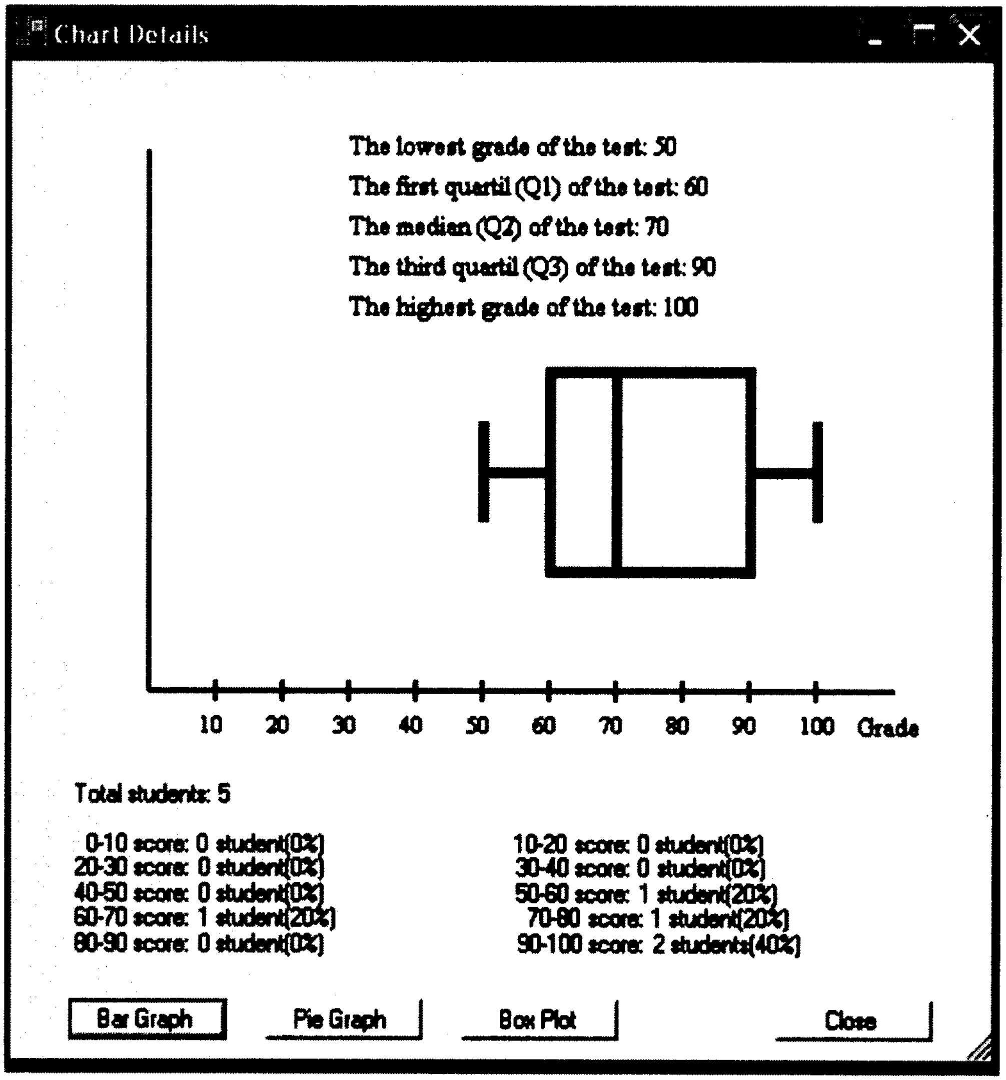
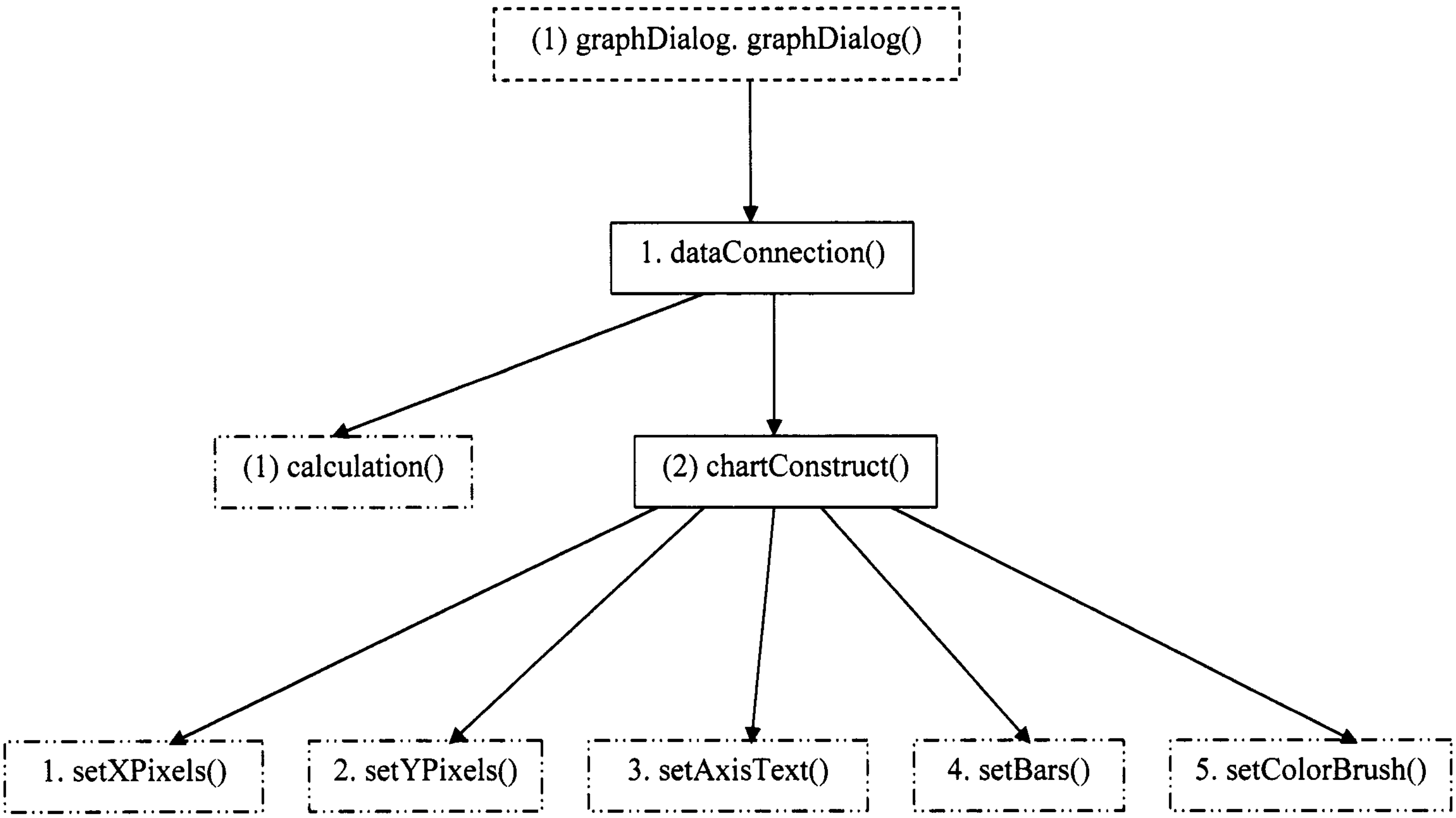


Figure 20: Box Plot.

Figure 21: Graph Dialog Control Flow.



6.3.2 Graphic Library

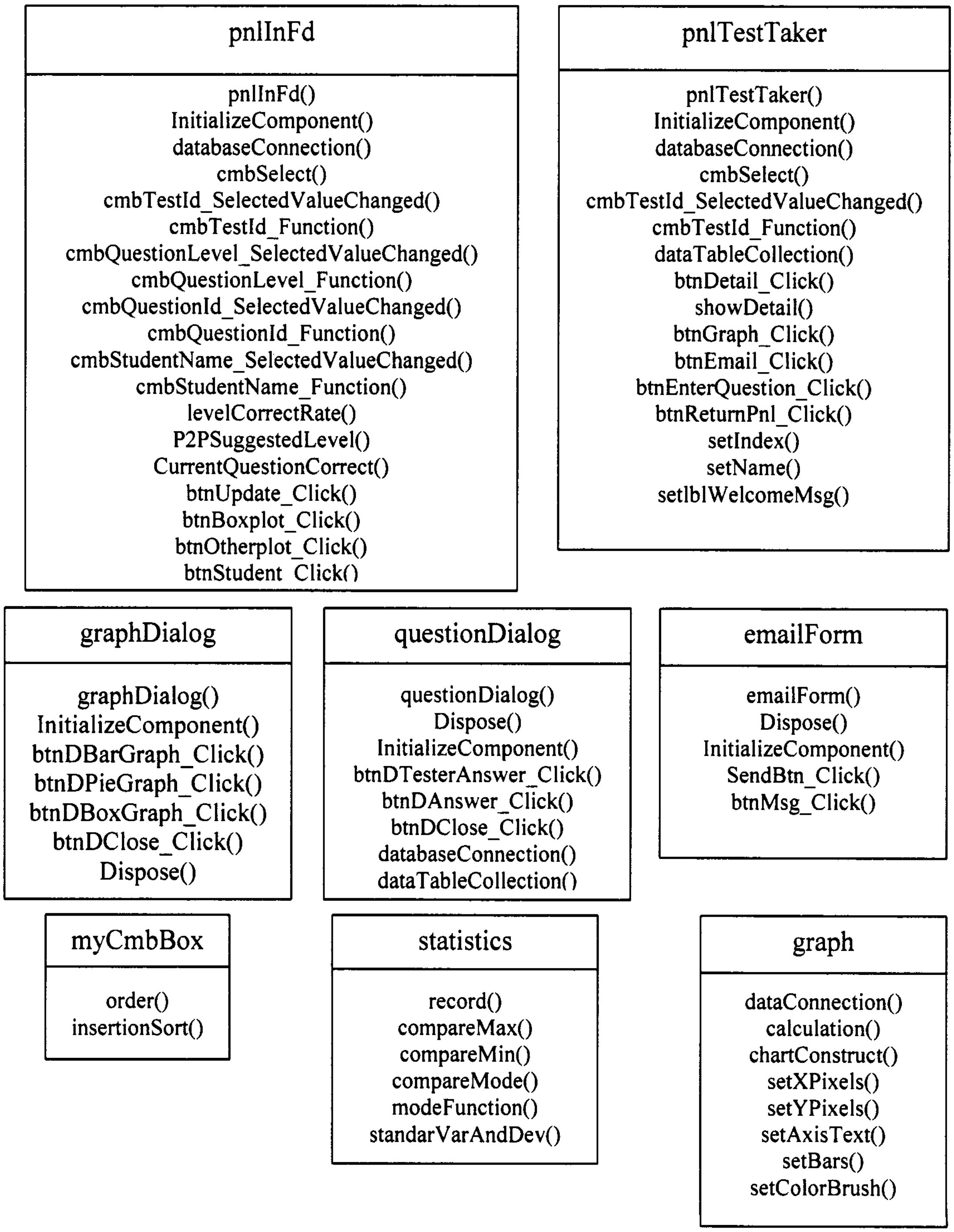
The presentation layer uses the Graphics, Pen, Bitmap, and SolidBrush classes of the .NET framework for drawing objects to be displayed in the graph dialog. All these classes belong to the System.Drawing namespace. The graphic class provides a rich set of graphic members, which can be used to draw different shapes such as lines, polygons, ellipses, etc. A pen object is used to draw a line of specified width and style. Brushes are used to fill shapes with a single color. A bitmap is used with images defined by the pixel data. Below is a code snapshot for using the graphic library.

```
private void chartConstruct(float[] gradeArry,
                    System.Windows.Forms.PictureBox pictureBox1,
                    string typeGraph,
                    double q_1,
                    double q_2,
                    double q_3,
                    int lowestGrade, int highestGrade)
{
      pictureBox1.BackColor =
             System.Drawing.SystemColors.ControlLightLight;
      myBitmap = new Bitmap(440,330); // size of pictureBox1
      g = Graphics.FromImage(myBitmap);
      myPen = new Pen(Color.Black);
      myPen.Width = 5;
      myBrush = new SolidBrush(Color.Red);

      // drawing x-axis
      g.DrawLine(new Pen(new SolidBrush(Color.Black),2),
                 40,myBitmap.Height-40, myBitmap.Width-20,
                 myBitmap.Height-40);


      // drawing y-axis
      g.DrawLine(new Pen(new SolidBrush(Color.Black),2),
                 40,myBitmap.Height-40, 40, 20);


      setXPixels(g, myBitmap.Width-60, myBitmap.Height-60,
                 40, myBitmap.Height-40);


      if(typeGraph == "barplot")
      {
            setYPixels(g, myBitmap.Width-60, myBitmap.Height-60,
                       40, myBitmap.Height-40);


            setAxisText(g, myBitmap.Width, myBitmap.Height,
                        "Grade", "Percentage(%)");


            // drawing bars
            setBars(g, myPen, myBitmap.Width-60,
                    myBitmap.Height-60, 40, myBitmap.Height-40,
                    gradeArry);

      }
}
```

6.4 Class Diagram

This section summarizes all the classes used by the feedback model with a brief description of the functionality of each class.

Figure 22: Class Diagram.



Class Description:

- Class *pnlInFd* describes handlers of the functions and the GUI of the instructor feedback.
- Class *pnlTestTaker* describes handlers of the functions and the GUI of the test taker feedback.
- Class *questionDialog* deals with the functions and GUI of the pop-up question dialog.
- Class *graphDialog* shows statistics for a particular test in graphic form.
- Class *emailForm* responds to the functions and GUI for sending email.
- Class *myCmbBox* ensures that all combo boxes are in number and letter order by providing an insertion sort.
- Class *statistics* provides all kinds of statistical calculations, such as calculating average, mode, rank, etc.
- Class *graph* provides common functions and GUIs to the three different kinds of graphs: the "bar plot," the "pie," and the "box plot."

CHAPTER 7
IMPLEMENTATION DETAILS AND DESCRIPTION

7.1 The Question Difficulty Level Feedback

7.1.1 Difficulty Level

To implement the question difficulty level feedback, five question difficulty levels are defined. The higher the level is, the harder the question would be. The percentage (X) of the correct answers is calculated by taking the total number of students who answered this question correctly and dividing by the total number of students who answered this question and multiplying by a hundred percent, (the number of students who answered correctly/ the total number students) * 100%. The percentages (A) of the correct answer for each of the other questions of the same level and the percentages (B) of the correct answers for all the questions with one level beneath and one above the level under consideration are calculated in the same manner. The mean of A and the mean of B are then calculated.

If X <= Average (mean(A), mean(B)) then the level under consideration is accurately assigned. The model will thus suggest to instructors that this question is at the correct difficulty level. This feature assists instructors in predicting the difficulty level of an individual question in any test and gives them dynamic suggestions to adjust this level if necessary.

Below is the rule of the relationship between the correct answered rate X of a question with the difficulty level under consideration and the suggested question level:

Correct Answered Rate	Suggested Question Level
X >= Average (mean of the percentages of correct answers at level 1, mean of the percentages of correct answers at level 2)	The first level (level 1)
X < Average (mean of the percentages of correct answers at level 1, mean of the percentages of correct answers at level 2), and X >= Average (mean of the percentages of correct answers at level 2, mean of the percentages of correct answers at level 3)	The second level (level 2)
X < Average (mean of the percentages of correct answers at level 2, mean of the percentages of correct answers at level 3), and X >= Average (mean of the percentages of correct answers at level 3, mean of the percentages of correct answers at level 4)	The third level (level 3)
X < Average (mean of the percentages of correct answers at level 3, mean of the percentages of correct answers at level 4), and X >= Average (mean of the percentages of correct answers at level 4, mean of the percentages of correct answers at level 5)	The fourth level (level 4)
X < Average (mean of the percentages of correct answers at level 4, mean of the percentages of correct answers at level 5)	The fifth level (level 5)

7.1.2 Error Checking and Error Messages

The model catches the errors below and provides the corresponding error messages.

- If users do not enter anything into the update text box and click the "Update" button, the program will display an error message, "Error! Please enter question level!" It is required that users enter the question level before they click the "Update" button to update the difficulty level of this particular question in the database.

- If users enter anything than a number for updating the question level, the program will display an error message, "Error! Question level must be a number!" because the difficulty level must be a number.
- If users enter a negative number for updating the question level, the program will show an error message, "Error! Question level must be positive!" because it is impossible for a question level to be negative.
- If users enter more than one number for updating the question level, the program will display an error message, "Error! Question level is a one-digit number!" This message notifies users that they can enter only a single digit number for updating the difficulty level.
- If users enter a number that is more than one digit for updating the question level, the program will display an error message, "Error! Question level is a one-digit number!" because there are currently only five one-digit difficulty levels for all the questions in the database.
- If users enter a positive number, which is less than one or greater than five for updating the question level, the program will show an error message, "Error! Question level must be between 1and 5!" because there are only five difficulty levels for all the questions in the database currently.

7.1.3 Warning Messages

The program pops up warning messages to users in the following situations:

- If instructors enter a different question level than the level the model suggests, a warning dialog will pop up to catch the instructors' attention so that instructors can have a chance to correct their input or confirm their decisions.
- If instructors enter the same difficulty level as the original one, a message, "You entered the same question level!" will show up and the program will not do anything.

7.1.4 Correct Input

If users enter correct input (1-5) into the update text box and click the "Update" button, the program will update the difficulty level for that particular question in the database. In addition, it will update the feedback GUI at the same time. In some situations when users enter a "+" notation before a number, the program is also able to recognize this positive number and to deal with this number by following the positive number rules.

7.2 The Statistical Feedback

7.2.1 General Statistical Feedback

7.2.1.1 Statistical Significance

In order to let the users know their positions in this specified test, the program provides statistical data and graphs in the statistical feedback. These statistical data are:

- All students' grades in this specified test
- The average of the test grades
- The standard variance of this test

- The standard deviation of this test
- The test grade range
- The mode of these test grades, which is the grade scale that most students got on this test
- The highest grade of this test
- The lowest grade of this test

From these general statistical data, instructors can easily see the students' performances for this specified test.

7.2.1.2 The Graph Field

The program also provides three different kinds of graphs to make it easier for the users to understand the data of this test. These three graphs are the bar graph, the pie graph, and the box plot. By clicking the "Box Plot" and the "Other Plots" buttons, users can choose the graph that they want to see from the pop-up "Chart Details" dialog. The default graph of the "Box plot" button is a box plot, and the default graph of the "Other Plots" button is a bar graph. Users can also change the default to other graphs by clicking the buttons in the "Char Details" dialog.

When users click the "Box Plot" and the "Other Plots" button, a "Chart Details" dialog will pop up. There are three parts in the "Chart Details" dialog: a graph field, some statistics, and four buttons – a "Bar Graph" button, a "Pie Graph" button, a "Box Plot" button, and a "Close" button.

The graph field shows the specified graph that is drawn based on the statistics of this test. There are three different graphs that users can choose to view from the graph field – the bar graph, the pie graph, and the box plot.

- Bar Graph: The bar graph is a two dimensional graph. The x-axis contains ten grade scales. The y-axis represents the percentage rate of each ten-point scale. There will be further explanation about this percentage rate and ten-point scales in "the statistical numbers" part. Based on the statistical numbers of these specified tests, the program will dynamically draw different color bars to represent each statistical result.
- Pie Graph: The pie (circle) consists of different color portions. Different color portions represent different ten-point scales. The percentage rate of every ten-point scale is displayed at the bottom right of the pie graph dialog.
- Box Plot: The box plot is one of the most important statistics graphs and contains some valuable statistical numbers. There is only an x-axis in the box plot. This x-axis contains ten grade scales. Usually, the box plot has a rectangle and three vertical lines: one line is inside the rectangle, and two lines are outside the rectangle and at two different sides of it.

The rightmost vertical line of these two outside lines represents the highest score (the fourth quartile) of this particular test, and the leftmost vertical line of these two outside lines represents the lowest score of this chosen test. The only line inside the rectangle represents the median score (the second quartile) of this specified test, which is the middle score of all grades in this test.

The right side of the rectangle represents the third quartile of all grades, and the left side of the rectangle represents the first quartile. In statistics, there are a total of four quartiles. Each quartile is divided into twenty-five percentiles. The score at the first quartile represents the score that is higher than twenty-five percent of all grades in this test. The score at the second quartile (the median score) represents the score that is higher than fifty percent of all grades. The score at the third quartile represents the score that is higher than seventy-five percent of all grades. The score at the fourth quartile represents the score that is higher than or equal to one hundred percent of all grades, and it is also the highest score.

Some numerical details are shown at the middle of the box plot:

- The lowest score
- The first quartile score
- The median score (the second quartile score)
- The third quartile score
- The highest score.

7.2.1.2.1 The Statistical Numbers

The statistical numbers show the total number of students who took this test, the number of students in every scale, and the percentage of students' grades in every scale. Every ten-point is a scale, so there are a total of ten scales. The table below shows these ten scales:

0-10 score (It includes 0, but does not include 10)
10-20 score (It includes 10, but does not include 20)
20-30 score (It includes 20, but does not include 30)
30-40 score (It includes 30, but does not include 40)
40-50 score (It includes 40, but does not include 50)
50-60 score (It includes 50, but does not include 60)
60-70 score (It includes 60, but does not include 70)
70-80 score (It includes 70, but does not include 80)
80-90 score (It includes 80, but does not include 90)
90-100 score (It includes both 90 and 100)

These percentages are calculated by taking the number of students whose grades are in that particular scale and dividing by the total number of students who took this test and multiplying by 100%, (the number of students in that particular scale / the total number of students) * 100%. For example, if the grades in test number one are 100, 90, 90, 70, 60, and 50, then the total number of students is six. Since there are three people in the "90-100 score" scale, the percentage rate of the "90-100 score" scale is fifty percent, (3 / 6) * 100% = 50%. The table below shows all the statistical numbers of this example:

Total student: 6	
0-10 score: 0 student (0%)	10-20 score: 0 student (0%)
20-30 score: 0 student (0%)	30-40 score: 0 student (0%)
40-50 score: 0 student (0%)	50-60 score: 1 student (16.7%)
60-70 score: 1 student (16.7%)	70-80 score: 1 student (16.7%)
80-90 score: 0 student (0%)	90-100 score: 3 students (50%)

The program checks for the number of students in order to display the result. If there is zero or one student, the program will display "student" after the number. If there are two or more students, the P2P program will display "students."

7.2.1.2.2 The "Bar Graph," the "Pie Graph," the "Box Plot," and the "Close" Buttons

There are four buttons in the "Chart Details" dialog: the "Bar Graph," the "Pie Graph," the "Box Plot," and the "Close" buttons.

The graph field corresponds to three of these buttons: the "Bar Graph," the "Pie Graph," and the "Box Plot" buttons. For example, if users click the "Bar Graph" button, the graph field will show a respective bar plot for this particular test. The other two buttons work the same way.

If users click the "Close" button, the "Chart Details" dialog will be closed, and users can continue to see more information from the returning instructor feedback GUI. The "Chart Details" dialog provides users with a considerable amount of valuable information. Users can gather a general view about the test and decide what they should do. For instance, from the graph the instructors can learn valuable information about this test and the students' level by comparing it to the standard set by the test. If this difference varies, professors may either need to revise the course or require more effort from the students. Students can also find out what their own study situations in this test are so they can improve themselves.

7.2.2 The Student Detail Feedback

7.2.2.1 The Statistical Numbers

These statistical numbers are similar to the statistical numbers in the instructor feedback. Detailed explanations are given in the statistical feedback part at the instructor GUI.

7.2.2.2 The "Test Detail", "Graph", and "Send Email" Buttons

Two separate GUIs are implemented to provide this detailed feedback. The instructor feedback GUI provides instructors with details about any student. This feedback feature is located at the bottom part of the instructor feedback GUI.

First, it shows the names of all of the students who took this test. Then instructors can get the individual student's record by selecting the student's name. Once instructors have selected the student, they not only get the individual student's grade, but also access the individual student's feedback GUI, which contains more specific test information about this student.

In the individual student feedback GUI, it shows the statistical numbers, the three buttons – the "Test Detail" button, the "Graph" button, and the "Send Email" button, and the test details.

The test detail part of the student feedback GUI is invisible at the beginning. Users must click the "Test Detail" button to see all the test details which include:

- The date the test was taken
- The test due date
- The time the test was taken
- The limit of the test time
- The total number of questions in this test
- The number of questions that this student answered correctly
- The number of questions that this student answered incorrectly
- The indexes of the questions that this student answered correctly
- The indexes of the questions that this student answered incorrectly

In addition, the GUI contains a text box and a "Submit" button. Users can see the details of individual question by entering the index of the question and clicking the "Submit" button. More will be discussed at the interactive feedback.

By clicking the "Graph" button, the "Chart Details" dialog will pop up. The default graph is the bar graph. The detailed explanation of the "Chart Details" dialog is at the "Chart Details" dialog part of the instructor GUI.

By clicking the "Send Email" button, users can send email to others. This feature will be discussed furthermore at the interactive feedback.

7.2.3 The Differences between the Instructor and the Test Taker Feedback GUIs

Below are listed the differences between the instructor feedback GUI and the test taker feedback GUI:

- The instructor feedback GUI contains more feedback features such as updating the difficulty level of a question, viewing information about any student taking the course while the test taker feedback GUI contains only statistical and interactive feedback. It is because only professors are authorized to update the level of the questions and to access any student information.
- Instructors can go to the individual test taker feedback for the chosen student by clicking the "Student Details" button at the bottom right of the instructor feedback GUI. Once the instructors get in the test taker feedback GUI, they can get all the information about this individual student. In addition, the instructors can always return to the instructor

feedback GUI at any time by clicking the "Return" button, which is located at the bottom right of the test taker feedback. Therefore, the instructors have the freedom and the priority to switch between the instructor feedback GUI and the test taker feedback GUI at any time. However, there is no way that students can go to the instructor feedback GUI from the test taker feedback GUI. There is no "Return" button at the test taker feedback GUI if the user is a student. The reason is simple: students have no priority to access the instructor feedback GUI and no right to access other students' test details.

7.3 The Interactive Feedback

There are two features belonging to the interactive feedback: sending an email and reading individual question's details.

7.3.1 Sending an Email

By clicking the "Send Email" button at the test taker feedback GUI, a "Send Mail" dialog will pop up. There are four text boxes in this dialog:

- The sender's email address
- The receiver's email address
- The subject of this email
- The email context

The sender's and the receiver's email addresses must be entered in correct standard email address format, such as john@yahoo.com, otherwise, the program will catch them as errors.

By clicking the "Send" button of this "Send Mail" dialog, users can send their emails to others. Once the feedback GUI sends the email successfully, it will display a successful dialog to users and users can just close this dialog by clicking the "Close" button in this dialog.

If users change their minds and do not want to send emails any more, they can just click the close button ("X") at the right top of the "Send Mail" dialog or click the "Close" button, which will automatically close the "Send Mail" dialog.

The "Send Mail" dialog also provides a default message, which includes general details about this particular test. These details are:

- Test ID
- Test average
- Test range
- Highest grade
- Lowest grade
- The mode scores

Below is an example of the default message:

The Test Details:

Test ID: 9111
Test Average: 74
Test Range: 50
Highest Grade: 100
Lowest Grade: 50
The Mode Scores: 100 – 90

Users can simply click the "Default Message" button to get this information and don't need to type everything into the message box.

7.3.2 Question Details Feedback

The test taker feedback GUI also provides a function for users to review a particular question for that test. Users can review any test question of the test by entering the question number and clicking the "Submit" button. Once users click the "Submit" button, a "Question Details" dialog will pop up. This dialog includes the original question description, the five answer choices, and the three buttons – the "Test Taker Answer," the "Correct Answer," and the "Close" button.

By clicking the "Test Taker Answer" button and the "Correct Answer" button, users can easily check and compare the test taker's answer with the correct answer. Then, users can review where the test taker made mistakes and what the correct answer to this question is. After the users have finished with reviewing this question, they can click the "Close" button to close this "Question Details" dialog. If users want to review more questions, they can simply enter other question numbers.

CHAPTER 8
CONCLUSION

The feedback model described in this paper constitutes one of the essential components in the P2P based active e-learning system. The feedback model provides users with useful feedback that assists and improves their learning and teaching skills. The algorithm that the model employs to intelligently calculate the difficulty level of a question is self-corrective and can be validated empirically.

The three-layer architecture of the model makes the system modular and thus easy to maintain. The software platform of the model is built on a Microsoft .NET framework, which is a language-neutral component library and execution environment. The open architecture and language independence of the model facilitates the integration of future applications and enhancement to the existing model.

Though the current feedback model is primitive, it contains the three major feedback categories: the intelligent feedback, the statistical feedback, and the interactive feedback, that any useful feedback system should have. Therefore, the model can serve as a practical model for any educational feedback model.

The paper also proposes a revised data mining technique, the "K Nearest Neighbors with Weighted Selection," which combines common data mining techniques with mathematical calculation and logistic computation. Though the algorithm is only moderately applied in the implementation of updating the difficulty level of a question, this technique can be applied for other various intelligent predictions.

The extended algorithm of building an entire neural network based on weighted values from the "K nearest Neighbors with Weighted Selection" method is worth considering. The complexity in building a model based on this algorithm is compensated for by the value that this model offers to the academic world.

REFERENCES

Berry, Michael J. A., Gordan S. Linoff. Data Mining Techniques: for Marketing, Sales, and Customer Relationship Managements. Indianapolis: Wiley, 2004.

Berson, Alex, Stepen Smith, and Kurt Thearling. Building Data Mining Application for CRM. McGraw-Hill Osbone Media, 1999.

Chase, Warren and Fred Bown. General Statistics. New York: John Weley and Sons, Inc., 2000.

Clark, D. Face-to-Face with Peer-to-Peer Networking [Electronic version]. Computer, 34(1) 18-21, 2001.

Date, C.J. An Introduction to Database Systems. Addison-Wesley Publishing Company, 1995.

Galitz, Wilbert O. The Essential Guild to User Interface Design: An Introduction to GUI design principles and techniques. New York: Wiley Computer Pub., 1997.

Hamilton, B. ADO.NET Cookbook. O'Reilly Publication, 2003.

Hathaway, Patti. Giving and Receiving Feedback: Building Constructive Communication. Menlo Park, Calif: Crisp Publications, 1998.

McMahon, Richard Alan, and Sr. Sean Chase. MCSE SQL Server 2000 Database Design. Scottsdale AZ: Coriolis Group Books, 2001.

Mena, Jesus. Data Mining Your Website. Cboston: Digital Press, 1999.

".NET Framework." Microsoft Corporation. 2004 <http://www.msdn.microsoft.com/library>.

Microsoft Official Course. Introduction to Microsoft .NET Development. 2003.

Pyle, Dorian. Data Preparation for Data Mining. San Francisco, Calif: Morgan Kaufmanns Publishers, 1999.

Sharp, J. & Jagger J. Microsoft Visual C# .NET Step by Step. Redmond, Washington: Microsoft Press, 2003.

Stephens, K. Ryan, and Ronald R. Plew. Database Design. Indianapolis, Ind.: Sams Pub., 2001.

"600 Word Vocabulary Test." 13 July 2004 <http://www.geocities.com/satyr37/>.

Thuraisingham, Bhavani M. Data Mining: Technologies, Techniques, Tools, and Trends. Boca Raton: CRC Press, 1999.