An Open Framework for Smart and Personalized Distance Learning

Ruimin Shen[1] , Peng Han[1], Fan Yang[1], Qiang Yang[2], and Joshua Zhexue Huang[3]

[1] Department of Computer Science & Engineering
Shanghai Jiaotong University, Shanghai, China
{rmshen,phan,fyang}@mail.sjtu.edu.cn
[2] Department of Computer Science
Hong Kong University of Science and Technology, Hong Kong, China
qyang@cs.ust.hk
[3] E-Business Technology Institute
University of Hong Kong, Hong Kong, China
jhuang@eti.hku.hk

Abstract. Web based learning enables more students to have access to the distance-learning environment and provides students and teachers with unprecedented flexibility and convenience. However, the early experience of using this new learning means in China exposes a few problems. Among others, teachers accustomed to traditional teaching methods often find it difficult to put their courses online and some students, especially the adult students, find themselves overloaded with too much information. In this paper, we present an open framework to solve these two problems. This framework allows students to interact with an automated question answering system to get their answers. It enables teachers to analyze students learning patterns and organize the web-based contents efficiently. The framework is intelligent due to the data mining and case-based reasoning features, and user-friendly because of its personalized services to both teachers and students.

1 Introduction

As distance learning becomes one of the hotspots in network research and applications, many web-based education systems have been established. Two good examples are Virtual-U [1] and Web-CT [2]. To cover the entire spectrum of the learning process, these systems have implemented a number of fundamental components such as synchronous and asynchronous teaching systems, course-content delivery tools, polling and quiz modules, virtual workspaces for sharing resources, whiteboards, grade reporting systems, and assignment submission components. These research and commercial e-learning systems enable large groups of dispersed individuals to interact, collaborate and study on the Web.

As distance learning becomes popular, new demands for more advanced features increase. For example, to satisfy the requirements of multimedia-based courses, teachers need to spend a lot of time to learn course-creation tools. This proves difficult for the senior teachers who are accustomed to the traditional ways of

J. Fong et al. (Eds.): ICWL 2002, LNCS 2436, pp. 19-30, 2002.

teaching. Another issue is that both the number of students using the Web based learning environment and the flow of e-learning materials grow very fast. This creates a problem of information overload for both students and teachers. Demands for personalized services increase. We note that the existing web-based systems often do not provide sufficient support on such aspects as giving personalized services to each individual student and helping them find their desired courses for study and answers to their questions. This problem has a great impact on the quality of network-based education and has contributed largely to the students drop rate.

In this paper, we present an intelligent distance-learning environment, which is developed and used at the Network Education College of Shanghai Jiao Tong University. The motivation of our work is to build a new distance learning system that enables students to conduct online studies easily according to their own educational backgrounds, study habits and paces. We are particularly interested in providing solutions to the information overload problem and personalized service. In short, our efforts are dedicated to make teachers feel that "everything is easy" and make students feel that "everything is available" and "everyone is different". Our system is being used by thousands of adult students regularly in Shanghai, China. In the following, we present the framework with an emphasis on the issues of providing answers to students' questions, and making personalized recommendations to students. We discuss data mining and case based reasoning techniques to solve these problems.

2 Overview of the System Architecture

The system is composed of a real-time classroom, an EOD (Education on Demand) course centre, a CBIR (Content Based Indexing and Retrieval) search interface, a learning assistance center and a data analysis center. During a class session, all the data the lecturer and students need, including video, audio, handwriting materials and screen operations, are transmitted simultaneously to each student's desktop. In the meantime, all interactions are recorded and public materials are published on the web. After the class session, students who were unable to take the class can view the same content on the web as that shown at the class. The CBIR search interface enables the students to find their desired materials conveniently and quickly. The learning assistance center consists of an assignment subsystem, an examination subsystem and an answer-machine subsystem that help students to complete assignments and exams on the Web, and answer their questions automatically. All the didactical and user access data are collected in log files and analyzed by the data analysis center. The system can provide personalized service to the students according to the analysis results. The details of these components are discussed in the following sections.

2.1 The "Everything Is Easy" Teaching Environment

Although multimedia tools have been built to help teachers create online courseware, some teachers still prefer to use blackboards. Especially, teachers teaching mathematics and chemistry feel it difficult to write complex symbols and formulas on

computer screens. To make "everything easy" for these teachers, we have developed an intelligent board transfer system. The teachers can write anything on a computerized whiteboard and the content is transferred simultaneously to the students' desktops and integrated with the teachers' video and audio teaching materials. The students can write notes on the teachers' handwriting window. The combined information is stored on the network so the students can review it anytime later. We called such content as personalized notes. The teachers can also load their pre-prepared PowerPoint and Word documents into the transfer system, and then both the teachers and students can navigate these documents synchronously. Using this subsystem, the teacher can focus on the teaching content instead of formats.

All the useful data from a class session are stored and published on the Web. The students missing the class session can learn themselves anytime after the class. We also convert these contents to CDs for the students who are unable to view the active online lessons due to the limited bandwidths.

Under such an environment the teachers and students can always find a time to communicate that suits for their work and preference. This conforms to our philosophy of "everything is easy".

2.2 The "Everything Is Available" Assistance Tool

A distance-learning environment often contains too many materials for students to choose. It is important to provide a tool for students to find the right materials they need. A lot of work has been done in the past on this aspect. However, many efforts have been put on standardizing the courseware with a unified data specification such as XML so that they can be indexed on the Web. We believe that it is even more important to design an interface for the students to decide whether the knowledge he is searching for is inside the courseware and locate it. For example, if a student wants to review "The First Law of Thermodynamics", he can input the phrase through a textbox or microphone, and then the computer can locate the relevant materials in the courseware automatically through an answer machine system and a speech recognition system.

In our system, we use a Content based Information Retrieval technology to implement this function. As we described above, the courseware includes such information as the teacher's video, audio and tutorials. We consider the audio and tutorial information to be the most important materials and index them. The students can see both the teacher's video and the didactical materials such as the PowerPoint slides, as shown in Figure 1. They can also hear the teacher's voice. In addition, the system can support the courseware on-demand with the index keyword input.

Fig. 1. Courseware on-demand based on CBIR (Content Based Information Retrieval)

Because the number of students is large, usually ten times or more than a conventional teaching class, a lot of teaching tasks have to be supported by the computer. Let's take Q&A(Question and Answer) System as an example. If there are 200 students online and each student asks only one question, then it will take a teacher several hours to answer all these questions. From our experience, many questions although expressed differently usually have the same or similar meanings. The solution to this problem is to share the answers among the students and let a computer recognize similar questions and answer them automatically. If the computer cannot find an answer, it transfers the question to a teacher. After the teacher answers the question, the answer is added to the Q&A database and shared among students. Therefore, as the Q&A database accumulates questions and answers, the hit rate grows over time.

There are already some existing question-answering systems in use. In comparison, our system emphasizes on efficiency rather than comprehension of the language. We have observed that only a limited number of questions are asked in each course and the questions are usually very simple. Thus, we adopt an improved key words matching algorithm to find the answer. After a period of accumulation, the hit rate of our Q&A system has risen to 90% and the corresponding time to answer each question is reduced to two seconds.

We first discuss the structure of our answer machine system in detail. The questions and answers are obtained through a standard web interface. The students using the system will leave behind many questions and potential answers. Over time, these questions and answers will accumulate in a log file. The log file can then be used for training an indexing structure for the question to answer association. This process continues whenever the system is in use, making the answer machine system a closed-loop system. We will adopt the lifetime learning paradigm of Yang and Zhang in [12] for acquiring indexical knowledge about cases in a case-based reasoning paradigm. In this paradigm, the answers are cases to be stored in a case base. The questions

provide keywords that trigger the cases and rank them according to how well they can provide an answer for the questions. An important issue then is how to provide ranking for the keyword to answer association. We call this the index-learning problem.

The structure of a case base can be conceptualized as a two-layer structure, where the feature-values form one layer and the cases another. The feature-value layer is connected to the case layer through a set of weights to be maintained. We now extend the original two-layer structure of a case base into a three-layer structure, taking the two-layer architecture as a special case. In the case layer, we extract the answers from each case, and put them onto a third layer. This makes it possible for different questions to share a solution, and for a question to have access to alternative answers. An important motivation for this separation of a structure of a case is to reduce the redundancy in the case base. Given N questions and M solutions, a case base of size $N*M$ is now reduced to one with size $N+M$. This approach eases the scale-up question and helps make the case base maintenance problem easier, since when the need arises, each question and answer need be revised only once. In order to make this change possible, we introduce a second set of weights, which will be attached to the connections between cases and their possible solutions. This second set of weights represents how important an answer is to a particular question if this answer is a potential candidate.

The weights correspond to a mapping function between the input questions and the final answers. Different questions may in fact correspond to the same answer. When many students ask questions, over time this mapping can be learned by a relevance feedback algorithm. We adopt the relevance-feedback learning algorithm proposed by Zhang and Yang [12] for case-based reasoning system, where the weights are incrementally updated based on whether a particular case provides a right answer or not for an input question.

In order to validate the system, we have to gather more data from the students. The data should not only reflect what questions the students asked, as in the search engine query logs, but also how they rank the returned results. Given these question-answer log files, we can apply the above learning algorithm and keep the question to answer mapping always current [12, 15].

2.2 The "Everyone Is Different" Personalized Service

In a traditional education system, the course content is static and the teacher's assignments given to different students are the same. In reality, students have different backgrounds and the knowledge structure is dynamic. Given such diversity, how do we analyze students learning behaviors, characteristics and knowledge structures? Furthermore, how do we send the feedbacks of learning states to teachers? In addition, how do we visualize the analysis results to teachers and students intelligibly? In order to answer these questions, we propose a subsystem ---the Data Analysis Centre, which includes an analysis tool to support the student study behavior analysis. Figure 2 gives the framework of the subsystem.

In this subsystem, the resource database is composed of two kinds of data: the log files with specification of W3C and the attribute tables in the sub-function database.

The data-preprocessing module will deal with the original data to clean them up. The first task is to transfer the log files into database files with DTS (Data Transformation Services) tools. The second task is to create the corresponding tables of User_ID and IP. The transformation also solves the problem of the one-to-many relation between student's User_ID and IP attributes. The third task is to calculate the click-time and browse-span of one URL, which is very important to mine the data structure of students. The last task is to create new tables and views for further analyses.

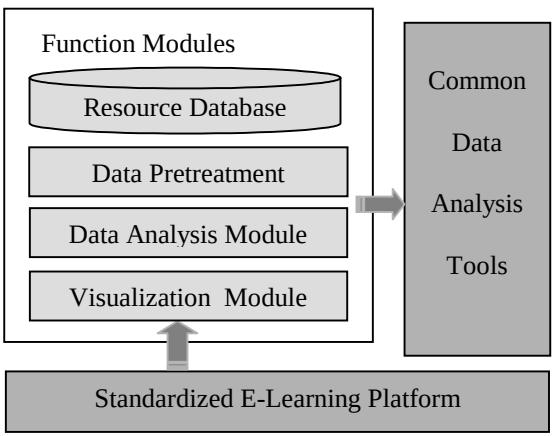

Fig. 2. Framework of the Data Analysis Centre

The preprocessing creates clean data. Since we organize data sources according to knowledge points and build relation tables of sources and knowledge points, we can assess the knowledge points from two aspects, the general information: to calculate the Interest Measure and the Mastery Measure of each chapter-point and knowledge-point based on the statistical data, and the personalized information: to assign the Interest Measure and the Mastery Measure to each student.

We use three techniques to discover knowledge and rules. The first technique is to use a classification algorithm to classify students into different classes based on their learning actions. Based on the classification, the teacher can organize different course contents and assign homework in different difficulty levels to each class. The second is to find association rules of different knowledge-points, the support and confidence values. The third is to organize and map the knowledge points using a concept map algorithm.

Using a visualization module, we can visualize all the analysis results in different forms. Figure 3-a shows the interestingness measure of knowledge points, based on the visit frequency of a certain chapter in a course, or the number of questions posted on the answer machine. It also shows the students' mastery measure of a given subject, determined by the students' feedbacks on whether they find the material satisfactory or not. The teacher can provide more scientific explanations online about a particular knowledge point with a high interestingness measure. He can also choose the low mastery measure knowledge point to teach in details and supply more reference materials to the students.

Figure 3-b shows the multidimensional association of knowledge points. The ellipses represent knowledge point groups, such as chapters. The circle represents a knowledge point. We can see not only the relationship between the knowledge points in the groups but also the relationship between the knowledge points in different groups. Such information can direct the teacher to re-organize the knowledge points more effectively.

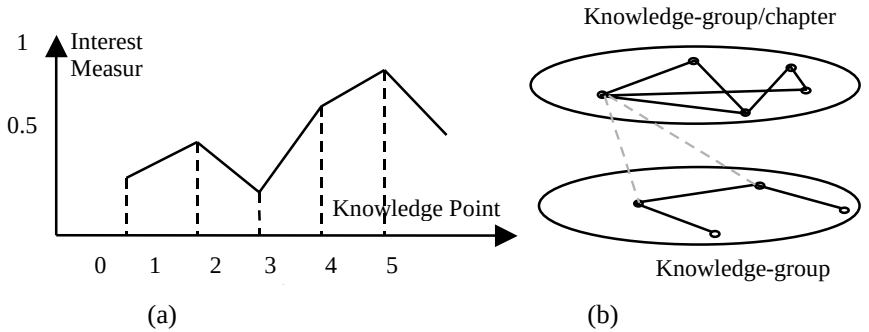

Fig. 3. Visualization of the Analysis Results

Furthermore, we can also represent a knowledge-point map which can show the relationship between the knowledge points and provide hints for the student as to what the prerequisite knowledge points are before the current knowledge point.

In our tests, the data Analysis Center can find some interesting rules and create useful graphs of the knowledge point structure. These results enable teacher to adjust the didactical progress and enable students to learn more personally.

Once we obtain the knowledge points, we now consider how to utilize the web log data accumulated by the web servers to derive interesting and useful association rules on the interesting knowledge points. Given a web log, the first step is to clean the raw data. We filter out documents that are not requested directly by users. These are image requests in the log that are retrieved automatically after accessing requests to a document containing links to these files. Their existence will not help us to do the comparison among all the different methods. We consider web log data as a sequence of distinct web pages, where subsequences, such as user sessions can be observed by unusually long gaps between consecutive requests. For example, assume that the web log consists of the following user visit sequence: (A (by user 1), B (by user 2), C (by user 2), D (by user 3), E (by user 1)) (we use "(…)" to denote a sequence of web accesses in this paper). This sequence can be divided into user sessions according to IP address: Session 1 (by user 1): (A, E); Session 2 (by user 2): (B, C); Session 3 (by user 3): (D), where each user session corresponds to a user IP address. In deciding on the boundary of the sessions, we studied the time interval distribution of successive accesses by all users, and used a constant large gap in time interval as indicators of a new session.

To capture the sequential and time-limited nature of prediction, we define two windows. The first one is called antecedent window, which holds all visited pages within a given number of user requests and up to a current instant in time. A second window, called the consequent window, holds all future visited pages within a number of user requests from the current time instant. In subsequent discussions, we

will refer to the antecedent window as W1, and the consequent window as W2. Intuitively, a certain pattern of web pages already occurring in an antecedent window could be used to determine which documents are going to occur in the consequent window.

The moving windows define a table in which data mining can occur. Each row of the table corresponds to the URL's captured by each pair of moving windows. The number of columns in the table corresponds to the sizes of the moving windows. This table will be referred to as the Log Table, which represents all sessions in the web log. Figure 4 shows an example of such a table corresponding to the sequence (A, B, C, A, C, D, G), where the size of W1 is three and the size of W2 is two. In this table, under W1, A1, A2 and A3 denote the locations of the last three objects requested in the antecedent window, and P1 and P2 are the two objects in the consequent window.

W1			**W2**	
A1	**A2**	**A3**	**P1**	**P2**
A	B	C	A	C
B	C	A	C	D
C	A	C	D	G

Fig. 4. A portion of the Log Table extracted by a moving window pair of size [2, 2]

We now discuss how to extract sequential association rules of the form LHS →RHS from the session table. Here LHS refers to the left-hand-side of a rule, whereas RHS the right-hand-side of a rule. The association rules have been a main subject of study in data mining [3, 4, 5, 6, 8 9]. Our different methods below will extract rules based on different criteria for selecting the LHS. In this work, we restrict the RHS in the following way. Let {U1, U2, …Un} be the candidate URL's for the RHS that can be predicted based on the same LHS. We build a rule LHS→ Uk where the pair {LHS, Uk} occurs most frequently in the rows of the table among all Ui's in the set {U1, U2, …Un}. Ties are broken arbitrarily. This is the rule with the highest support among all LHS→Ui rules.

The first rule representation we consider is called the subset rules. These rules are the same as the traditional association rules which simply ignore the order and adjacency between accesses. Thus, when the association rule mining methods, such as the Aprioi method [4,5,6], are applied to the log table, we obtain the subset rules. As an example, table 1 shows the subset rules that can be extracted from a transaction.

Table 1. Subset rules

W1	W2	Subset Rules
A, B, C	D	{A, B, C} → D, {A, B} → D, {B, C} → D {A, C} → D, {A} → D, {B} → D, {C} → D

The second rule representation is called the subsequence rules, which takes into account the order information in the sessions. A subsequence within the antecedent window is formed by a series of URLs that appear in the same sequential order as

they were accessed in the web log data set. However, they do not have to occur right next to each other, nor are they required to end with the antecedent window. When this type of rules is extracted from the log tables, the left hand side of the rules will include the order information. Table 2 shows an example of subsequence rules.

Table 2. Subsequence rules

W1	W2	Subsequence Rules
A, B, C	D	(A, B, C) → D, (A, B) → D, (B, C) → D (A, C) → D, (A) → D, (B) → D, (C) → D

For each rule of the form LHS→ RHS, we define the support and confidence as follows:

$$\sup = \frac{count(LHS, RHS)}{count(Table)} \quad (1)$$

$$conf = \frac{\sup(LHS, RHS)}{\sup(LHS)} \quad (2)$$

In the equations above, the function count(Table) returns the number of rows in the log table, and

$$\sup(LHS) = \frac{count(LHS)}{count(Table)} \quad (3)$$

From these rules, we can obtain interesting association relations between courses. For example, our rules can inform the teachers "Students who find Chapter 3 useful also find Chapter 5 useful". Knowledge like this will allow the teachers to organize the two chapters together on the web structure. It will also allow teachers to recommend students new chapters to read based on their current reading. Similarly, the same associations can be used to help organize the material better or form better student study groups. For example, a rule such as "Students who attends classes in the Wednesday classes often have difficulty with Calculus I" enables the teacher to improve the Calculus I material better online, or organize the students in that class to work together with students from other classes. We also plan to use different users information and log data to perform collaborative filtering analysis and provide recommendations [7] using Pearson Correlation.

The above-discussed framework assumes that the knowledge points are given beforehand. However, these knowledge points can be discovered from the web logs as well. Pitkow J. and Pirolli P. in [10] provide a longest subsequence mining method for extracting user profiles. Su, Yang, Zhang, Xu, Hu and Ma [14] provide an interesting method for clustering based on the web logs alone. In our study, we plan to combine both the content information and the user behavior information from the web logs to derive the clusters. The method that we propose to use is called clustering. Due to space limitations, we will not go into details on this subject.

3 A Distance-Learning Case Study

When a student connects to our NEC (Network Education College) home page (http://www.nec.sjtu.edu.cn), he can select which chapter or section to study. Our system provides multimedia study materials for students, including video, audio, images and text documents. The learning resources are well organized for study convenience. During a student's learning session, he may have a question to ask. Our system provides a functional button in every study page to help the student link to the Answer Machine at any time. When the student clicks the "Answer Center" button, he can see the Ask Question page. In this window, he can input the question in natural language and submit it as shown in Figure 5.

After receiving this initial query, the system shows a list of similar questions to the student. The student can choose the most similar one to see the answer. If all listed questions are not relevant, the student can submit the question to a teacher (see Figure 6). Beyond these functions, the Answer Center also provides other services, such as the Hot Spot of Lesson, the Hot Spot of Chapter, Search Answer and so on. For example, the Hot Spot of Chapter can provide the hotspots discussions of every chapter. The hotspots discussion can help students to find out what questions other students have asked and what the correct answers are.

The user can see the distribution of questions of a chapter or section in the selected time-span. The results can be shown in graphs, pie charts, histograms and so on. The user can choose different forms he likes and looks into details by clicking each part of the diagram (see Figure 7).

In addition, the relation of knowledge points can be shown in 2D or 3D graphs. According to the precedence and subsequence of a knowledge point, the system can recommend the imperative knowledge to learn or to prepare.

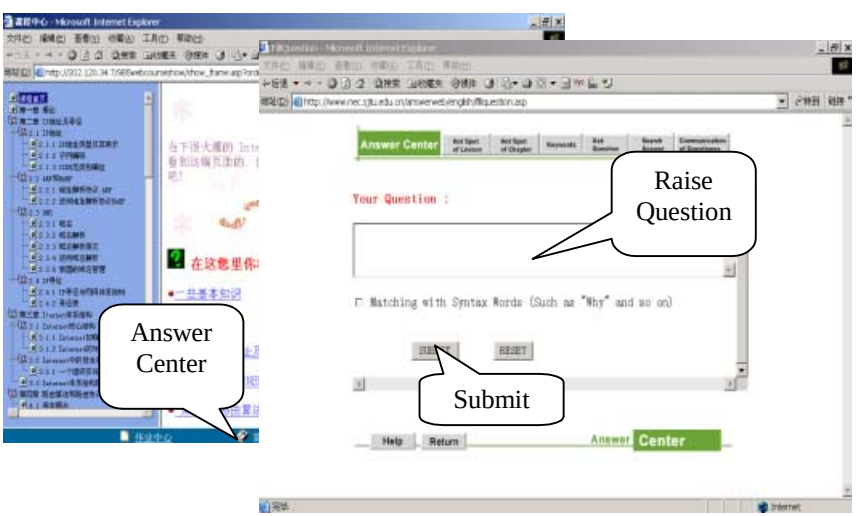

Fig. 5. Learning and Submitting Questions

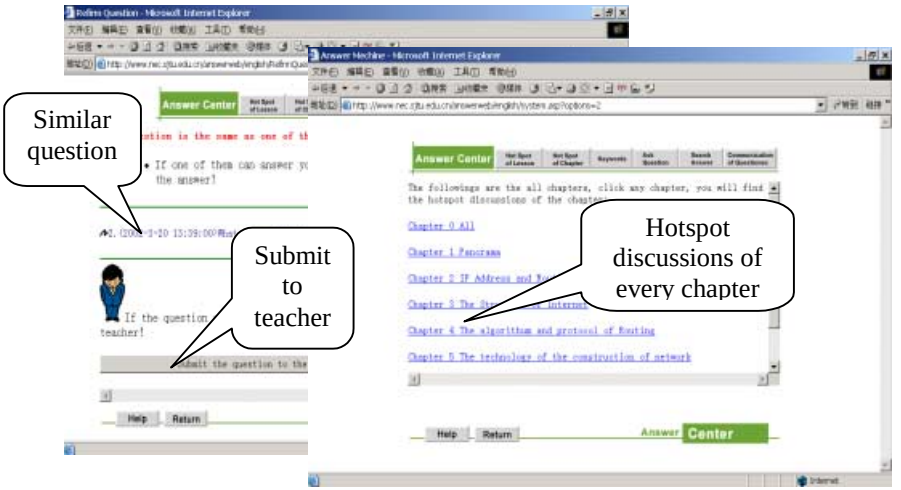

Fig. 6. Answering the questions in Answer Centre

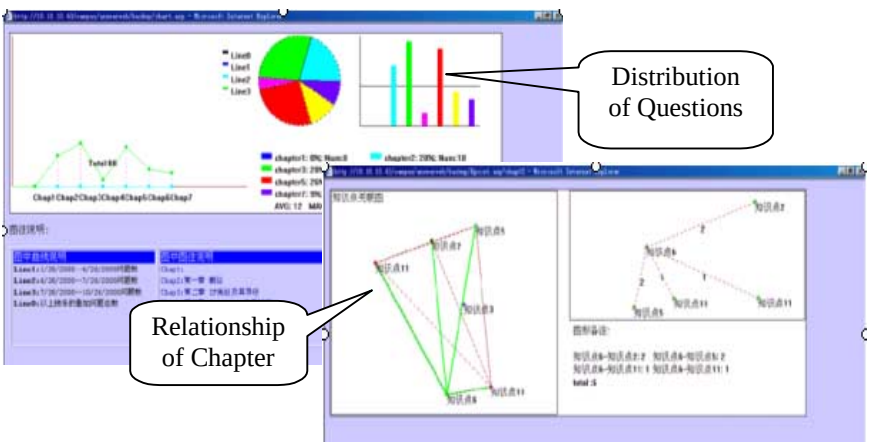

Fig. 7. Visualization results of the data analysis center

4 Conclusions and Future Work

In this paper, we have presented an open, adaptive framework to organize the course material. The heart of the intelligent system lies in a smart front end system we call Answer Machine, and an Intelligent back end system using web log association analysis and clustering analysis. In the future, we plan to offer more tests on the

systems performance using the data we accumulate through real teaching sessions. Such validation will allow us to select the best intelligent teaching methods for an open virtual teaching environment.

References

1. C. Groeneboer, D. Stockley, T. Calvert: Virtual-U: A collaborative model for online learning environments, In: Proceedings of the Second International Conference on Computer Support for Collaborative Learning, Toronto, Ontario, December 1997.
2. WebCT: http://www.webct.com
3. R. Agrawal and R. Srikant: Fast algorithms for mining association rules. In: Proceedings of VLDB'94, Santiago, Chile (1994), 487-499.
4. J. Han and Y. Fu: Discovery of multiple-level association rules from large databases. In: Proceedings of VLDB'95, Zürich, Switzerland (1995), 420-431.
5. R. Srikant and R. Agrawal: Mining generalized association rules. In: Proceedings of VLDB'95, Zürich, Switzerland (1995), 407-419.
6. R. Srikant and R. Agrawal: Mining quantitative association rules in large relational tables. In: Proceedings of SIGMOD'96, Montreal, Canada (1996), 1-12.
7. J. Breeze, D. Heckerman, and C. Kadie: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in AI, Madison, WI. (1998).
8. S. Chee, J. Han, and K. Wang. RecTree: An Efficient Collaborative Filtering Method. In: Proceedings of the DaWaK 2001, 141-151.
9. Q. Yang, H. Zhang, and I.T. Li: Mining Web Logs for Prediction Models in WWW Caching and Prefetching. In: Proceeding of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'01, Industry Applications Track, San Francisco, California, USA (2001), 473—478.
10. J. Pitkow and P. Pirolli: Mining Longest Repeating Subsequences to Predict WWW Surfing. In: Proceedings of the USENIX Annual Technical Conference(1999).
11. V. Ganti, J. Gehrke, and R. Ramakrishnan: Mining very large databases. COMPUTER, Vol. 32, No. 8, (1999), 38-45.
12. Z. Zhang and Q. Yang: Feature Weight Maintenance in Case Bases Using Introspective Learning. Journal of Intelligent Information Systems, Vol. 16, Kluwer Academic Publishers (2001), 95-116.
13. I.T. Li, Q. Yang, and K. Wang: Classification Pruning for Web-request Prediction. In: Poster Proceedings of the 10th World Wide Web Conference (WWW10), Hong Kong, China (2001).
14. Z. Su, Q. Yang, H.J. Zhang, X. Xu, Y. Hu, and S. Ma: Correlation-based Web-Document Clustering for Web Interface Design. International Journal of Knowledge and Information Systems. (2002) 4:141-167.
15. Q. Yang and J. Wu: Enhancing the Effectiveness of Interactive Case-Based Reasoning with Clustering and Decision Forests. Applied Intelligence Journal, Vol 14. No. 1., Kluwer Academic Publishers (2001), 49-64.