

WebCAT*—A Web-Centric, Multi-Server, Computer-Assisted Testing System

Ying-Dar Lin

ydlin@cis.nctu.edu.tw

Department of Computer and Information Science

Chien Chou

cchou@cc.nctu.edu.tw

Institute of Communication Studies

Yuan-Cheng Lai

Wen-Chung Wu

Department of Computer and Information Science

National Chiao Tung University

1001 Ta-Hsueh Road

Hsinchu, Taiwan

This paper describes the WebCAT* system. The WebCAT* is a Web-centric, multi-server system that combines the WWW, database and Internet: all test items are stored in databases located at each CAT server; users can access these materials via the Internet through a Web browser. The testbases of these CAT servers can be shared among each other in a way unseen to end users. Each test item in the WebCAT* is associated with degree of difficulty, domain, and the testbase it belongs to. Upon test generation, a test specification with test domain and max/min degree of difficulty is given to generate the test. Each test consists of items taken automatically, at random, from all the available testbases according to the specifications. This system has three major characteristics: multi-testbase sharing, real-time test generation, and item cross-analysis. A case study of an English exam with text and audio test items is given at the end of this work as an example. The overall feasibility of network-based testing is discussed as well.

INTRODUCTION

Along with advances in computer network technology, the Internet is getting more and more popular nowadays. Numerous topics can be found through the Internet: business information, advertisements, news and so on. Among all of them, distance learning -- education in a virtual classroom -- is one of the most practical and potential applications on the Internet (e.g., Squires, 1991; Jacobson & Levin, 1993; Barnard & Sandberg, 1993; Lollar, 1995). Along with this growth of distance learning is a new need for distance testing, hence this study.

Several distance learning environments are currently available on the Internet. Some of them are listed in Table 1 with short descriptions. One major characteristic of these systems is the interactivity among the participants so they can discuss with each other and their teachers.

Insert Table 1 here

Assessment is an essential aspect of all learning experiences. Tests play an important role in helping instructors evaluate student performance, and students play a role too. Some Computer-assisted Testing (CAT) systems, as shown in Table 2, have been developed to help students assess their learning. However these systems are stand-alone and specific to one particular subject domain. Students cannot access the test materials through the networks, thus these systems are not particularly convenient. Therefore, a more general CAT system which can be utilized via the network is desired.

Insert Table 2 here

The main idea of the networked CAT system is that all the test items are stored in a database on a server. Teachers can add new items on the client side into the database through an interface provided by the system, and students can take an exam in which the

contents are generated at random by the system. When the exam is finished, the grade can be calculated and shown to the student immediately and is recorded into the database. Both teachers and students can view these records later to check what subjects require more attention.

The primary factor affecting the effectiveness and fairness of the test is the number of test items in the database; therefore, a multi-server CAT system—WebCAT*—is developed in the present study to enrich the number of test items. WebCAT* is a Web-centric system in that all the interfaces the teacher/student interacts with are through a Web browser. Besides, the testbase of each server is shared with other servers, which enlarges the pool of available test items tremendously. In addition, cross-analyzing the test results can measure both the degrees of difficulty for individual test items and evaluate the entire testbases. The “*” highlights system’s feature of multiple cooperative servers involved.

The paper first presents main ideas and design issues, and then the implementation details and algorithms. The third part of this paper discusses a case study of a sample English exam. Possible future work and conclusions are also provided.

DESIGN OF WebCAT*

Overview

Some terms used in this paper are explained here.

- *Test item*: Each individual question in the database is called a test item.
- *Testbase*: All test items belonging to the same subject constitute a testbase.
- *Test profile*: A test profile is used to identify the scope of an exam; the WebCAT* system will then generate an “exam paper” according to the parameters in the test profile.
- *Test union*: A group of CAT servers that share testbases among each other.

The general structure of the WebCAT* system is presented in Figure 1.

Insert Figure 1 here

Each CAT Server can work independently. Users interact with it through a Web browser. To extend the available test items, these stand-alone servers can work together to establish a Test Union. All the testbases are then shared among each other in a way transparent to the end users.

Each CAT Server contains a database and has a daemon running on it. The database is composed of the test items, personal data, test records and so on. The daemon is a Web server process. A Web server is a computer with several running processes, with the fundamental process being an HTTP daemon. The HTTP daemon process handles HTTP requests to access hypertext, or other data and applications. WebCAT* is built by mounting various CGI and JAVA programs on the HTTP daemon. It is responsible for registering/deregistering when a CAT server wants to join/leave the Test Union.

Test items are inserted into the testbase via a Form which is then a tool used to gather information from Web users. Taking an exam in the WebCAT* system is simple: one selects an appropriate test profile first, then an “exam paper” will be generated automatically, according to the selected test profile. The grade is shown immediately after one submits the paper; some adjustment about the degrees of test item difficulty may also be performed accordingly. Details about the test generation method are given in the Section of Test Generation.

Three major characteristics are provided by the WebCAT* system:

- Testbase sharing: The number of usable test items for each organization in the Test Union is increased and the accuracy of test evaluation is upgraded.
- Real-time test generation: Different tests are generated each time, which prohibits cheating and makes network-generated testing more feasible.
- Cross-analysis of results: The degree of difficulty for each test item and the whole testbases can be measured more accurately.

System Structure

For each CAT Server to cooperate with each other, a TestBase Sharing Layer (TSL) is needed. The TSL should be transparent for users, and be between the Interactive Layer and Database (Figure 2). Three major functions are provided by TSL:

- Processing input data: All the data users' input is through the Interactive Layer, which is shown as a FORM on the browsers. These data are extracted and processed here first; then appropriate responses arranged in HTML are sent back to the Interactive Layer.
- Interacting with local database: New test items, student accounts, test records, etc., can be added or deleted from the local database. This is achieved using embedded SQL in this layer.
- Retrieving test items from other testbases: When generating a test, test items can be selected from other testbases if the CAT Server joins the Test Union. The test items should be retrieved at random and the item numbers should be proportional to the total test items each CAT Server possesses.

In order to reach the above goals, the Common Gateway Interface (CGI) programs were adopted in this study.

Insert Figure 2 here

Identity Classification

Three kinds of users are allowed to perform different operations. They are given differential privileges in order to protect databases from being destroyed or misused accidentally. The kinds of users and their roles are listed below:

- **Administrator:** Each CAT Server must have at least one administrator who acts as a “supervisor” of the system and who takes the responsibility for doing the following jobs:
 - (1) Join/Leave the Test Union: In order to use test items from other CAT Servers, the first thing to do is to join the Test Union. Leaving the Test Union at any time for maintenance or other reasons is considered legal and valid.
 - (2) Create/Delete accounts: Adding new user accounts to the system is the main job

an administrator must do. This will create a login name and an initial password for each user at the local database. When old accounts are no longer used, they will be removed from the local database to keep it updated.

- (3) Establish new testbases: A CAT Server might contain test items for different subjects; therefore, a testbase is used to classify different subjects and their subclasses. Which testbase a test item belongs to is specified when the administrator adds it into the database. Other jobs an administrator has includes creating new testbases or appending new subclasses to existing testbases.

- **Teacher:** Adding test items is the primary job a teacher should do; in addition, specifying the scope of an exam is also required so that correct test items can be retrieved from the CAT Servers when a student wants to take a test.

- (1) Add test items: To enlarge the available test items, all the teachers should add new questions into the database frequently. Each item should contain the correct answer, proper testbase it belongs to, and so on.

- (2) Make a test profile: The scope of an exam is specified by the teacher and is recorded as a test profile into the database. It should include the subject of the test, total test items needed, test allowed time, and so on.

- **Student:**

- (1) Take an exam: Before starting an exam, the appropriate test profile must be selected first. Test items are then selected at random from the WebCAT* system before the exam begins. Exam results will be recorded into the database after the exam is finished.

- (2) Practice: A student can make his own test profile to simulate an exam, or just select one subject and do the exercise items one by one.

An identity check is performed each time a user wants to enter the system. In addition to the functions described above, any legal users can view the statistical results of the exam to understand the performance of the overall group.

Registration/Deregistration

Each CAT Server in the WebCAT* systems must have the information of the current available members so it can get new test items from them. One way to do this is to register at a fixed site when a new server wants to join the Test Union. On the other hand, notification of leaving members is also required to keep everyone updated.

The responsibility for accepting a JOIN/LEAVE request and notifying others belongs to the Registration Server; this server functions as a normal CAT Server except it can also handle the registration of new members and the deregistration (removal) of outgoing ones.

Three phases are introduced during a registration process (Figure 3). Each phase is described below:

- (1) *JOIN request*: The new CAT Server sends the Registration Server a JOIN message with its own IP address, database name, etc. The Registration Server then adds it to its local database.
- (2) *Acknowledge (ACK)*: An ACK message listing the IP address of all members of the Test Union is sent back by the Registration Server to the new CAT Server. The newly-joined CAT Server updates its local database and becomes part of the Test Union.
- (3) *Broadcast*: The Registration Server informs other members of this newly-joined Cat Server by broadcasting a notification message to all members of the Test Union. Each member then updates its database and can start using the test items in the new CAT Server. At this point, the WebCAT* system is in consistent state and can operate correctly.

Insert Figure 3 here

The deregistration or removal process works in a similar way to the registration process, except that the message exchanged is different from the previous one:

- (1) *LEAVE request*: The leaving CAT Server sends a LEAVE message to the Registration

Server. The Registration Server will delete it from its local database.

- (2) *Acknowledge (ACK)*: An ACK message is sent back to the leaving CAT Server. It then deletes the information about the Test Union from its database and become a standalone CAT Server.
- (3) *Broadcast*: The Registration Server informs other members of this event and each member updates its database accordingly. The WebCAT* system is in consistent state again.

Test Generation

A “test paper” is generated at random each time when a user wants to take an exam. Some parameters designated in the test profile, such as subject, subclass, number of questions in each subclass, and degrees of test item difficulty are needed when producing a test paper. Four steps required during the generation process (Figure 4) are explained below.

Insert Figure 4 here

- (1) *Collecting qualified test items*: The number of qualified test items are queried from all the available CAT servers. The process continues to the next step if the total number of qualified test items is greater than the required number; otherwise, an error message is returned and the program ends.
- (2) *Deciding the item numbers each server should provide*: The number of test items selected from each CAT server should be proportional to the number of qualified test items it possesses. This is accomplished by generating a series of numbers ranging from 1 to the total number of qualified test items at random. The more test items a CAT server has, the more numbers will fall within its range, which is the amount of test items that will be retrieved from it.
- (3) *Selecting test items from each server*: Test items are now selected from each CAT server with the specified number at random again. A complete “test paper” is then produced at the end of this step.
- (4) *Printing-out the test paper in HTML*: The generated test paper has to be rearranged in

HTML so it can be shown on Web browsers. Adding HTML tags into the test paper is therefore required and is done in this step. The resulting “test paper” is now sent to the student, and he or she can now start the exam.

Database

The database plays an important role in the WebCAT* system: all the test items, personal records, and other information are stored here. Each CAT Server has its own database and only those that contain test items are shared with others; local information such as test records are not accessible by others. Therefore, the database used in the WebCAT* system can be roughly divided into two categories:

- Local: User accounts, test profiles and test records can be accessed by local users with appropriate privileges only. This keeps others from modifying the local data and is thus safer.
- Sharable: Test items can be selected each time a test paper is generated by one of the CAT Servers. By doing so, more usable test items are available and the network-based testing becomes more feasible.

IMPLEMENTATION

Overview

As indicated above in Figure 1, the WebCAT* is a multi-server system that follows traditional client-server architecture. One major characteristic of the WebCAT* system is that all the programs and system setups are done on the server side; that is, only a Java-compatible Web browser is required at the client side, no matter what platform is being used. Interaction between server and client is done via the Form and processed using CGI programs. Four major components are included according to the identity of the users: administrator, teacher, student, and analyst. The functions of each component will be described in the Section of Major Component. The software structure of WebCAT* is illustrated in Figure 5.

Insert Figure 5 here

Development Environment

The WebCAT* system is developed in the Solaris 2.x environment. Each server must have an http daemon running on it since it is a Web-centric system. In addition, a database management system (DBMS) is also required on each server to store the information; Postgres95 (ref. Postgres95 Homepage), a DBMS developed at the University of California at Berkeley, was used in this study for its simplicity and ease of use. The Interactive Layer introduced in the Section of System Structure is composed of the Form, Java (Campione & Walrath, 1996) and JavaScript (Netscape Communications Corporation, 1995-96), and the TestBase Sharing Layer is implemented using CGI programs written entirely in Perl (ref. The Perl Language Homepage), a script language powerful for manipulating strings. In addition, some Perl modules are included in the programs:

- (1) CGI.pm (Stein, 1997): A Perl5 module was used to create Web fill-out Forms and parse their contents in a fairly easy way when writing CGI programs.
- (2) pg95perl5: The Perl5 version of Postgres95 API (Application Programming Interface).
- (3) GD.pm: A Perl5 graphic module that can be used to create color drawings using a large number of graphics primitives, and emit the drawings as GIF files.

System Setup

The first step to setup the WebCAT* system is to create the required database used in the WebCAT* system. A script file, *initial.pl*, written in perl, is executed to achieve this goal using the following syntax:

initial.pl <ip_address>

One administrator's account is created during the execution process; the administrator is then responsible for creating other accounts (students, teachers, or other administrators) and new testbases. Teachers can now input test items and create test profiles; students can take exam or do exercises by themselves. The overall work flow is depicted in Figure 6.

Insert Figure 6 here

A daemon, *server.pl*, has to be running at each server in order to register into/deregister from the Test Union. After this is done, the testbase within the Test Union can then be shared.

Major Components

Three major components corresponding to three different kinds of persons introduced in the Section of Identity Classification are included in the WebCAT* system. In addition, one more component, analysis, is also appended into the system to examine the performance of each individual test item or the overall CAT Server. Users have to choose one appropriate component first and then pass the identity check before starting the system. Implementation details are described in the following subsections.

- **Administrator**

The administrator is the only one account created when the CAT server is first setup, as stated in the Section of System Setup. The interface that he or she interacts with is illustrated in Figure 7. On the left hand side are the functions that an administrator can perform; the right hand side is the corresponding form when one function is selected.

Insert Figure 7 here

The “*Status Report*” function shown in Figure 7 tells the administrator the current status of the CAT server, including status (Stand-alone, Networked, or Register Server), available testbase, and the number for each person. The button “JOIN” in the upper right-hand corner, is used to join the test union when the system is in Stand-alone state; it will change to a “LEAVE” button if the server is in a Networked state.

Creating new accounts is a primary job for the administrator. Two approaches are provided when adding new accounts: personal data or group data. For person data, only

one account is created with the specified login name and password, which is quite suitable when creating a new administrator's or teacher's account. Group data can be used when a large number of accounts (for example, a whole class) should be generated at one time. The initial password for each account generated in this way is the same as the login name; users should later change their passwords for security reasons.

Since an account may become obsolete after a period of time, the “*Modify Account*” function is provided. Invalid accounts can be removed from the database or passwords can be changed if the user has forgotten his or her password.

New testbases should be created before any teacher can input test items. The name of the testbase and each subclass must be the same for each CAT server if they want to share the testbases among themselves. New subclasses can be added to the existing testbase; on the other hand, unused testbases/subclasses can also be removed from the database when there are no test items belonging to it. The “*View Testbase*” function is provided to achieve these goals.

● **Teacher**

The teacher is to input test items and make test profiles so the students can take an exam or do exercises. Such a test profile interface is illustrated in Figure 8.

Insert Figure 8 here

A test profile is used to identify the scope of an exam. Teachers have to choose the subject, subclass, and fill out the form to complete a test profile. The “total questions” and “total score” fields at the bottom of the form are generated automatically when a teacher has filled out the “subclass question” and “subclass score” fields.

After a test profile is saved into the database, a sample “test paper” can be generated according to the parameters set in the test profile. This is provided by the “*View Test*

Profile” function. The teacher can save this sample test paper into the database. Later, when a student wants to take an exam, the same test paper will be retrieved if the teacher selects this kind of test profile; otherwise, the test paper will be generated at random. This is quite useful if the teacher wants the same test paper to be taken by all the students.

Two types of test items are currently supported: Normal questions and Sound questions. The former is just the usual text question; users have to key in the test items one by one. The latter is the audio-type test items; teachers should record this kind of question to a single file in an .au format since this is the only format that Java currently supports. The interface used to edit test items is shown in Figure 9.

Insert Figure 9 here

In addition to the interface for input test items done one by one as shown in Figure 9, another kind of test item, group questions, is also available in the WebCAT* system. The “*Group Questions*” function is used to provide an interface that can edit group questions. Also “*View Questions*” lets one see the currently available test items in the local testbase; teachers can remove those test items that he provided.

- **Student**

Students may use the WebCAT* to practice the subject they are interested in or to take an exam that the teacher assigned. To take an exam, a teacher-designated test profile has to be selected first. After the appropriate test profile is selected, the WebCAT* system will generate an exam from the available testbases. An example is given in Figure 10. The exam will then be sent back to the CAT server automatically if the test time is up, or manually if the student presses the “Turn in” button. Results will be evaluated immediately and shown on the Web browser, as illustrated in Figure 11. Each exam the student has taken is recorded in the database and can be reviewed later using the “*Personal Record*” function. Both the correct and incorrect answers will be marked so that the student can identify what he or she missed.

Insert Figure 10 here

Insert Figure 11 here

The “*Practice*” function lets students practice by themselves. Two modes are currently available: simulate an exam sheet or give one by one items. In the simulated exam sheet, students can make their own test profiles and start the exams. Results will also be shown immediately but are not recorded into the database. In the one-by-one item mode, students have to select one subject/subclass first, then the test items belonging to the selected field will be retrieved from the local database at random, with one question shown at a time.

- **Analysis**

Any legal users can use the analysis component of the WebCAT* system, but with different privileges: a student can only view those exams that he has taken; teachers can examine the exam records of those who have taken the test profiles they assigned; and the administrators can review all the records currently available on the local database. For each selected test profile, a bar chart will be shown that indicates the performance of the overall group (Figure 12). For teachers and administrators, the “View Detail” button can be used to see the detailed record for the selected student.

Insert Figure 12 here

CASE STUDY

A practical example using the WebCAT* system is given in this section. Problems encountered and possible solutions are also discussed.

Test Contents and Environment

Three CAT servers are included in this trial: one is installed on a SUN SPARC 20 workstation with a Solaris 2.5 acting as the registration server; the other two are installed on two PCs with Intel Pentium 133 processors with Solaris x86 2.5 CPUs. Two CAT servers, the registration server and a common one, are located at National Chiao Tung University (NCTU) ; the other is located at National Tsin Hwa University (NTHU). Both universities are in Hsinchu, Taiwan. An English testbase is created at each server, with contents drawn from TOEFL test items. The two common servers have been registered at the registration server before any test trials started.

Students' accounts are all created at the registration server. End users from both NCTU and NTHU are connected to this one to start the test. The platform each user uses is Microsoft Windows 95 or Windows NT 4.0, and Netscape Navigator 3.0 is the Web browser chosen to communicate with the CAT server. A test profile (Figure 8) is used to generate the test paper; the number of test items is one-third of that of the TOEFL exam. Test results are illustrated in Figure 12.

One major difference between these trials and the usual exam is the ability to test the student's listening comprehension. An added convenience is that users from different sites can take the exam provided a browser is at hand. Furthermore, test items are taken at random from all the available CAT servers; this prohibits cheating and makes network-based test more feasible.

Lessons Learned

Under normal operations, the WebCAT* system can generate a test paper according to the specified profile. Some issues are observed from these trials:

- In a few cases, the sound question presented were intermittent when the user played it. Possible reasons may be that the platform the user works on is itself unstable, or perhaps have been too many stuff installed so that the Java applet can not work as expected.
- The hard disk at user's PC should have enough usable space left since the test items will be downloaded from the servers to the user's temporary directory before an exam

begins. The browser may go down if there is not enough memory; this usually happens when an exam contains too many sound questions.

- The browser's proxy option should be disabled since the test items are retrieved from each server at random; there is no benefit if it is enabled.

Discussion

The WebCAT* system can be used to take an exam as observed from the above trials. However, the situation becomes more complicated if it is used to evaluate the student's performance. The primary reason that makes network-based testing difficult is how to prevent cheating, including how to authenticate identity. Possible methods of network-based testing are illustrated in Table 3.

Insert Table 3 here

For exams requiring a fixed place, on-site supervisors are required to check id.; otherwise, it is difficult or even impossible to know if it is the correct student or not. In this case, the test result can only be treated as a reference and cannot be used as the student's real grade.

For fixed-place exams, two alternatives are available: the student can take the exam at the specified time, or at any time. Some important exams need to be held at the specified time and place using the same exam paper to help achieve fairness. Other exams such as certification exams, can be taken at any time but with different test items. Therefore, the number of test items is the main factor affecting the fairness of the exam. The WebCAT* system allows testbases to share among each other, which enlarges the available test item pool enormously, and is applicable to this kind of test.

CONCLUSIONS AND FUTURE WORK

A prototype of a WebCAT* system is developed in this study. It lets each CAT Server share testbases, generating test papers according to the specified test profile. Although it may not entirely replace the traditional paper-and-pencil or stand-alone computer-assisted

test, it provides an alternative for evaluating the performance of dispersed students. The practical use of the WebCAT* system nowadays may be for students to take mock exams by themselves or take exams under non-traditional circumstances. In either case, the result will be shown immediately so the user can know which part he or she missed.

Some enhancements can be added into the WebCAT* system as discussed below.

- *Refinement of user interface:* This study used Form and JavaScript for the system to interact with its user. In future studies, they may be replaced by more versatile interfaces such as Java applet or ActiveX controls[18].
- *Diversity of test items:* Two types of test items are currently available: the usual text type and the audio-type. Other kinds of test items, e.g. video-type, can also be included. Also, test items are now limited to single-choice answers. Other styles, e.g. multiple-choice or matching, can also be used.
- *Classification of test items:* In this system, one test item can belong to only one subclass of a testbase, which is not realistic. Therefore, it is suggested that subclasses of each testbase should extend to a hierarchy structure so test items can be classified more clearly.
- *Analysis of testbase:* Item analysis is very limited in the current prototype system. Parameters associated with each test item should be modified so that cross-analysis can be performed.
- *Improvement of test generation algorithms:* Parameters used to generate test papers are still limited. A more general algorithm with more parameters can be used to replace the current one.
- *Fault tolerance of Test Union:* The WebCAT* system can operate normally only when each CAT server operates correctly. If one server in the Test Union misfunctions, e.g. the server is down, faulty performance may occur when a CAT server is generating a test paper. This should be improved in the future.
- *Combination with distance learning:* The WebCAT* system can be accessed by using a Web browser; the browser should be associated with a Web-based distance learning

system so students can learn and be tested on the materials in a similar environment.

REFERENCES

- Barnard, Y., & Sandberg, J. (1993). Open learning environments: What support should they offer? *Proceeding of International Conference on Computers in Education* (pp. 156-161), Taipei, Taiwan, December 1993.
- Campione, M., & Walrath, K. (1996). The Java Tutorial. On-line available at: <http://java.sun.com/Series/Tutorial/>.
- Chung, A. (1992). Computer assisted testing of mastery knowledge with Chinese characters. *Communications on the Move*, Singapore, ICCS/ISITA, 3, 1992, pp. 1057-1061.
- Davis, J. L., & Smith, T. W. (1994). Computer-assisted distance learning, Part I: Audiographic teleconferencing, interactive satellite broadcasts, and technical Japanese instruction from the University of Wisconsin-Madison. *IEEE Transactions on Education*, 37(2), 228-233.
- Jacobson, M. J., & Levin, J. A. (1993). Hypertext and network-based learning environments: Technology for the construction of personal and shared knowledge spaces. *Proceedings of International Conference on computers in Education* (pp. 282-287), Taipei, Taiwan, December 1993.
- Lin, Y. D. (1997). CORAL*: A Multi-server distant cooperative learning system. *Proceedings of International Conference on Computer Assisted Instruction* (pp. 15-22). Taipei, Taiwan, March 1997.
- Lollar, R. B. (1995). Distance learning for non-traditional students to study, near home, toward a UNC Charlotte BSET degree. *Proceedings IEEE Southeastcon ' 95*, 366-367.
- Maly, K., Abdel-Wahab, H., Overstreet, C. M., Wild, J. C., Gupta, A. K., Youssef, A., Stoica, E., & Al-Shaer, E. S. (1997). Interactive distance learning over Intranets. *IEEE Internet Computing*, January-February, 60-71.
- Netscape Communications Corporation (1995-96). JavaScript authoring guide. On-line available at: <http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/>.
- Ng, J. M., Chan, E., & Tsang, P. H. H. (1994). Interactive group discussion system for distance education. *IEEE First International Conference on Multimedia Engineering Education* (pp. 270-275), 1994.
- Squires, D (1991). Learning through multi-user network CAL. *Computers and Education*, 16, (1), 87-91.
- Stein, L. (1997). CGI.pm-a Perl5 CGI Library, Version 2.31. On-line available at: http://www-genome.wi.mit.edu/ftp/pub/software/WWW/cgi_docs.html.
- Walworth, M., & Herrick, R. J. (1991). The use of computers for educational and testing purposes. *Proceedings Frontiers in Education Conference* (pp. 510-513), 1991.

Table 1

Some distance learning environments

System	Media	Description
MITS (Ng, Chan & Tsang (1994)	Internet	Text editor, image viewing system, audio channel and so on are provided to support group discussion.
Audiographics (Davis & Smith, 1994)	High-speed Modem over phone lines, or Satellite Broadcasts	Audio conferencing with still frame graphics; all sites share the audio and visual information equally.
IRI (Maly et al., 1997)	Intranet	Fully interactive (audio, video and tool sharing) system; class preparation tool and WebBook are provided to help teachers and students.
CORAL* (Lin, 1997)	Internet	Multi-server, Web-based system; chat room, white board, guided navigation system and so on are supported.

Table 2

Some CAT systems

System	Subject focus
CAT (Walworth & Herrick, 1991)	Introductory electrical and analog circuit.
CAT (Chung, 1992)	Mastery knowledge with Chinese characters.

Table 3

Possible methods of test-over-network

<i>Testing place</i>	<i>Testing time</i>	<i>Test paper</i>	<i>Open/closed book</i>
Fixed	Specified	Same one	Open or closed
Fixed	Non-specified	Individual	Open or closed
Non-fixed	Specified	Same one	Open book
Non-fixed	Non-specified	Individual	Open book

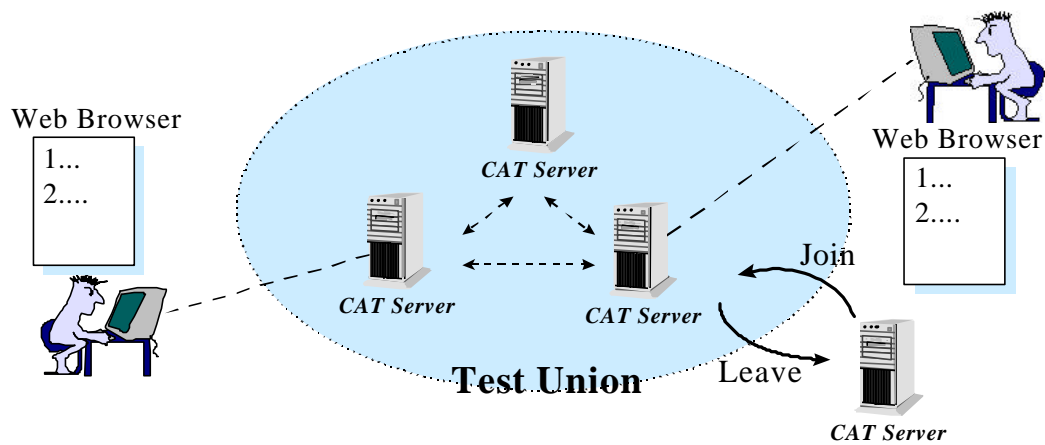


Figure 1. Structure of the WebCAT* system

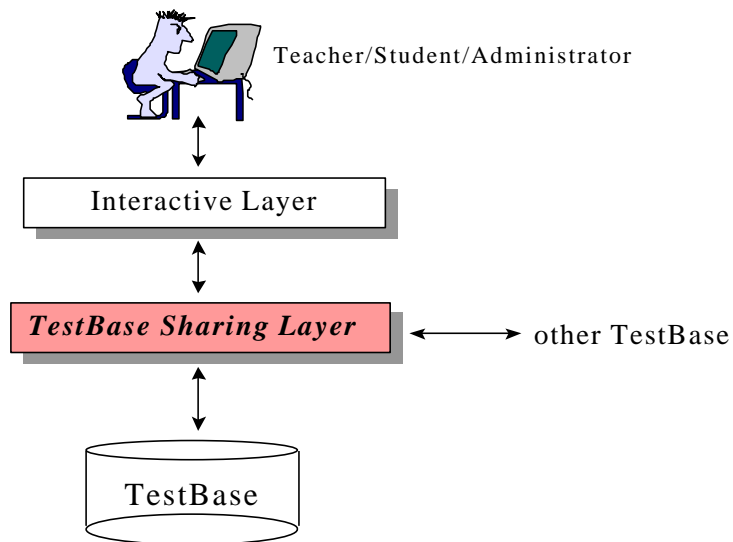


Figure 2. System structure of a CAT server

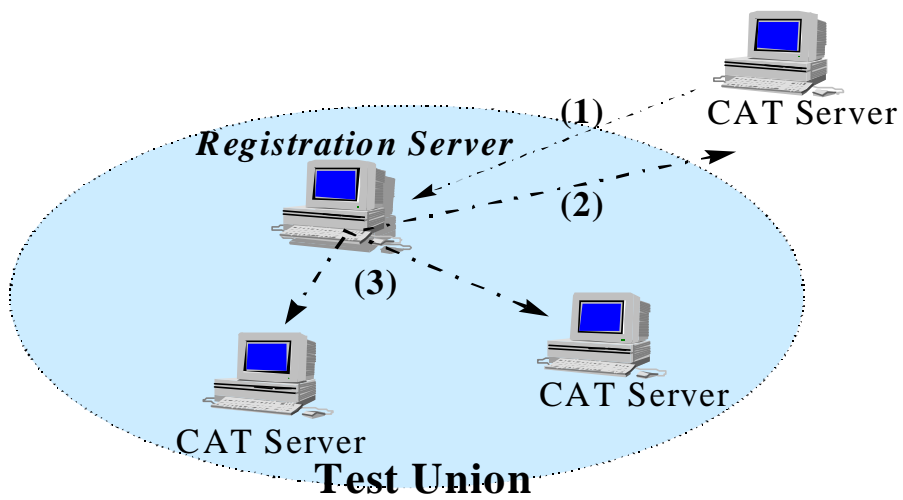


Figure 3. Registration/Deregistration process

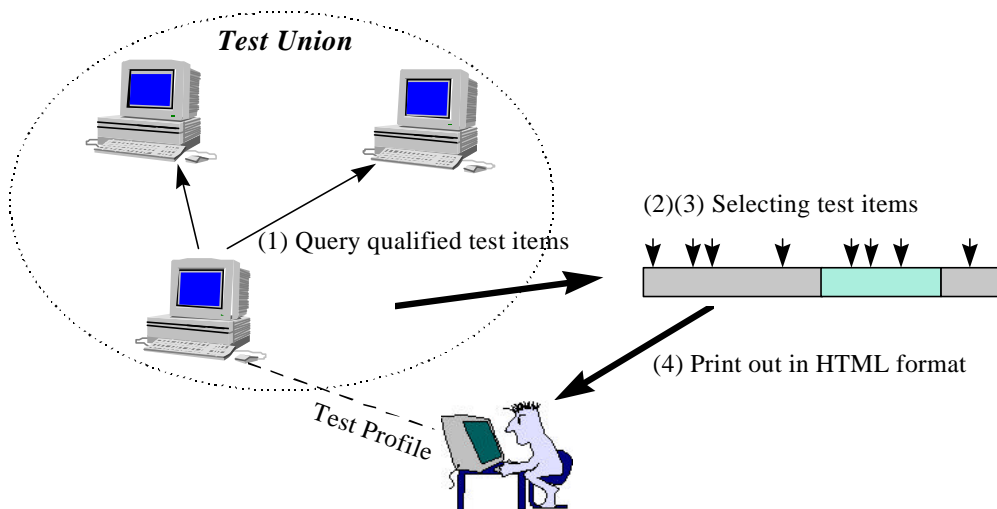


Figure 4. Test generation process

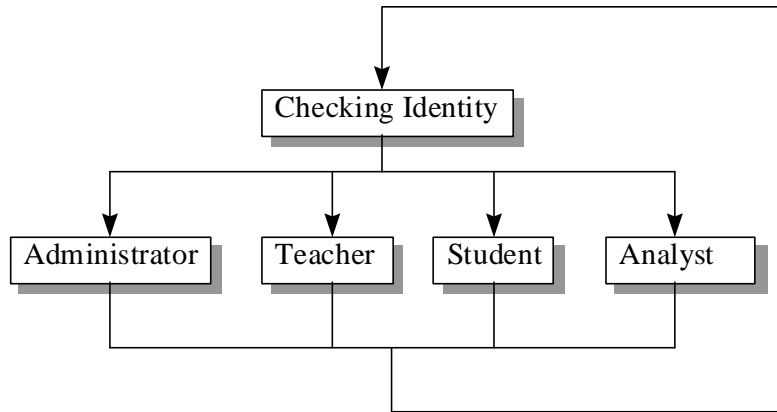


Figure 5. Software architecture of WebCAT*

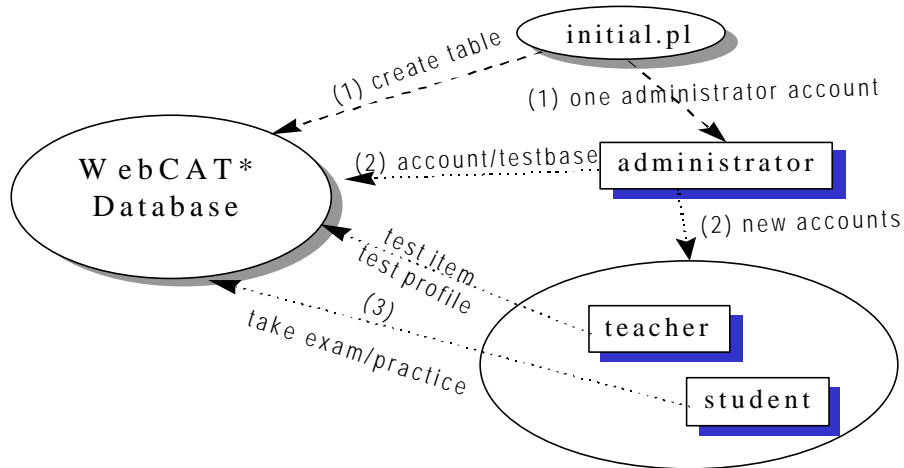


Figure 6. System setup procedure of the WebCAT* system

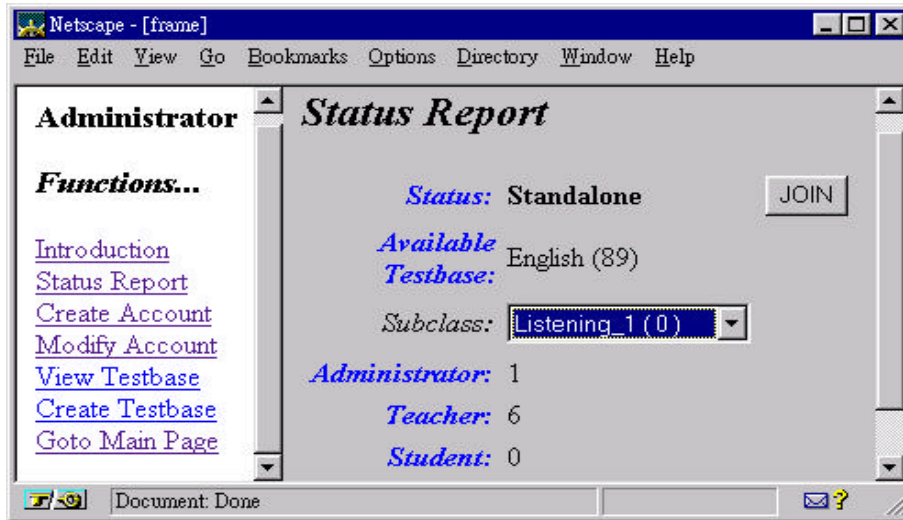


Figure 7. Current status of the CAT server

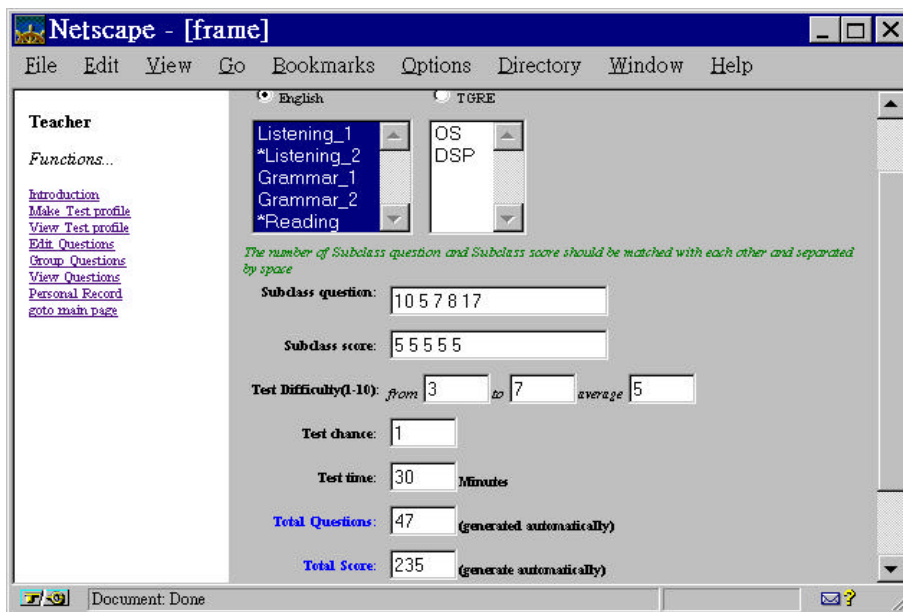


Figure 8. Make test profile interface

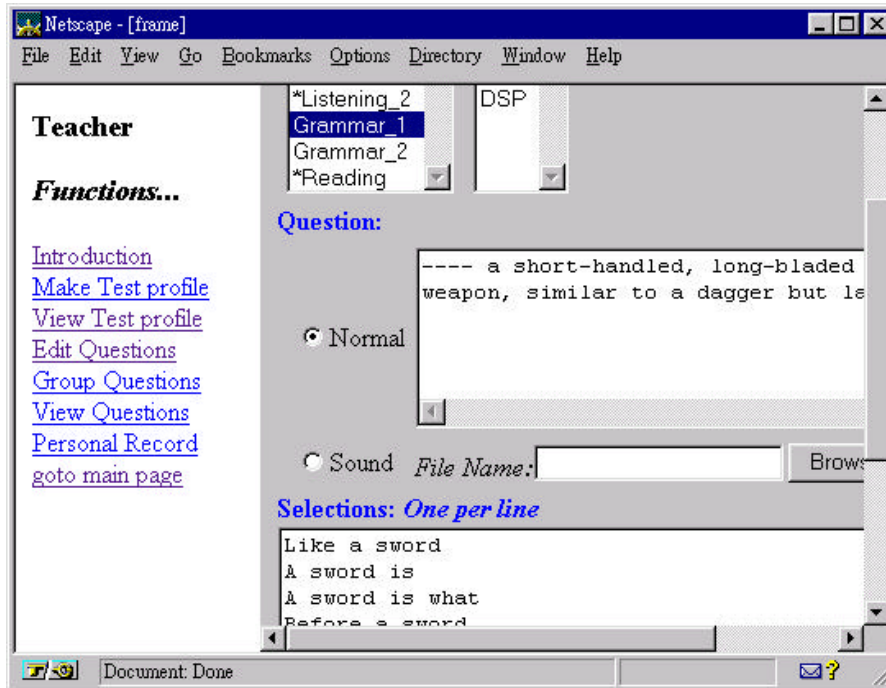


Figure 9. Edit test item interface

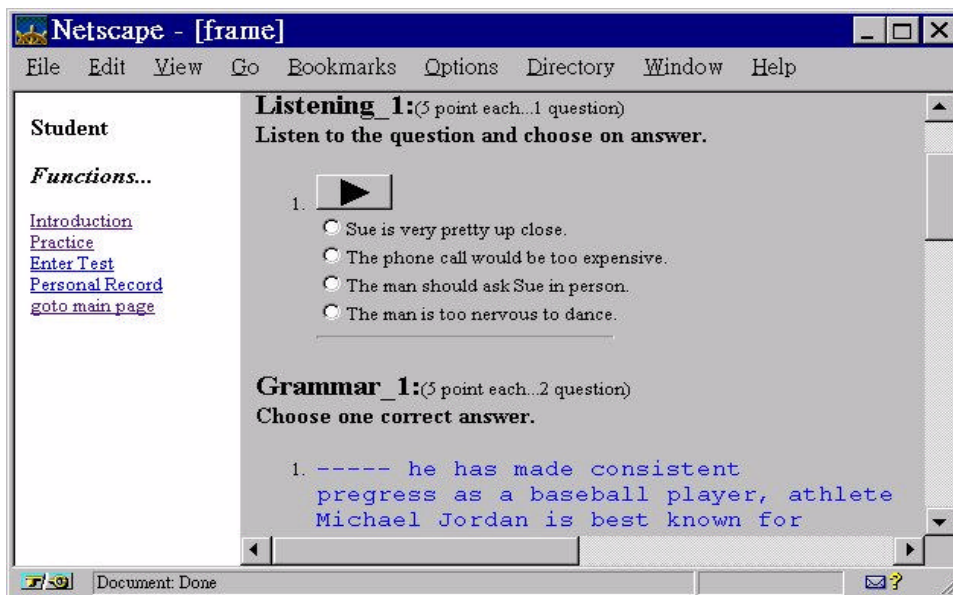


Figure 10. An exam paper generated from the WebCAT* system

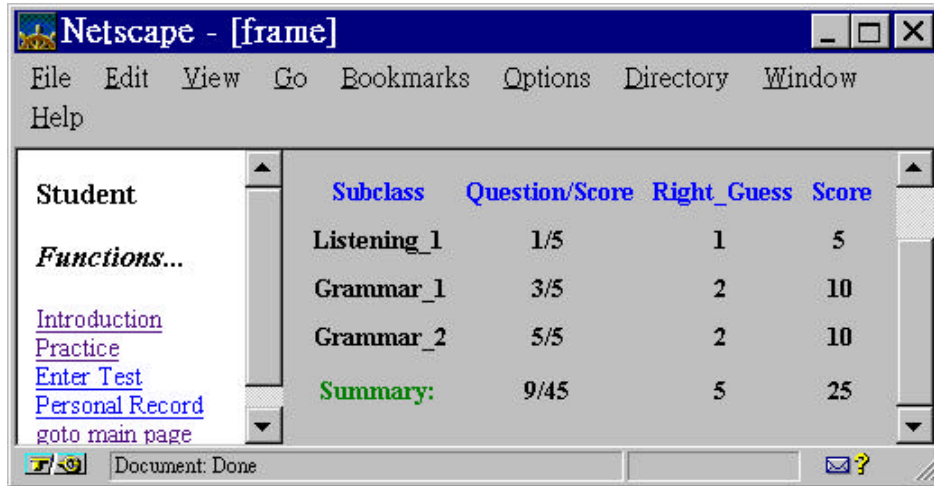


Figure 11. Exam result

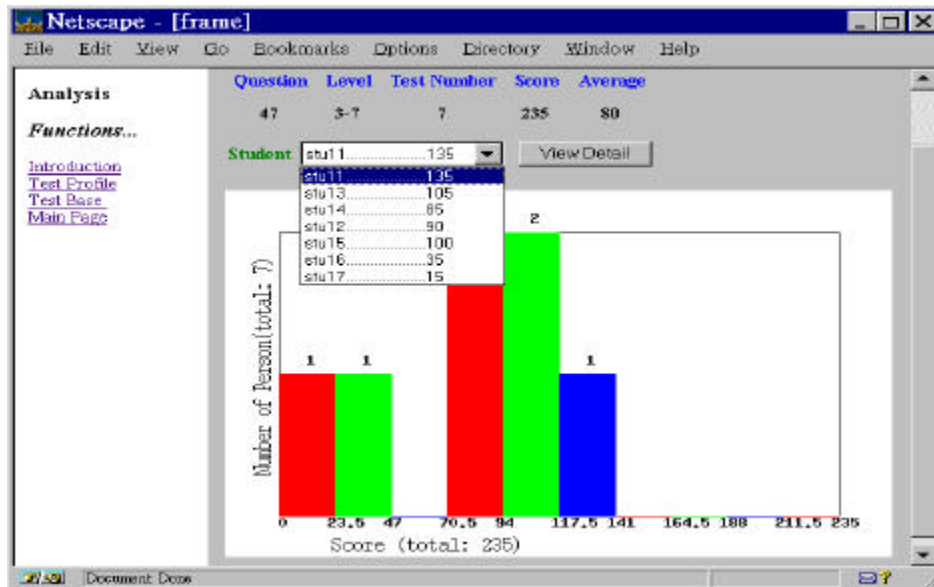


Figure 12. Bar chart showing exam results