Wireless Online Tutoring System (WOTS) With Collaborative Diagramming

Teddy C Y Chow, Edward W T Tsui, Stephen C F Chan & Vincent T Y Ng,
Hong Kong Polytechnic University, Hong Kong
E-mail: csschan@comp.polyu.edu.hk

Abstract

The wireless online tutoring system (WOTS) is a distributed tutoring system with high scalability and reliability. WOTS provides support for textual chat as well as interactive and collaborative structured-diagramming (UML). Users can access WOTS from both personal computers and handheld devices. Evaluation results show that the load-balancing mechanisms can evenly distribute system load among all active servers and reduce the mean process time of user requests.

1. Introduction

Web-based tutoring requires distributed, scalable and reliable web-based tutoring systems. In addition to text chatting, another fundamental form of communication is through structured diagrams, or concept maps. There are currently numerous web-based learning and tutoring systems. For example, NetChat 0 is a web-based communication and collaboration tool designed for educational purpose. It provides chat room, newsgroup and graphical whiteboard. However, NetChat provides only a simple whiteboard, but no support for complex conceptual diagrams, such as UML diagrams, which is very popular in computer science. There is also no mechanism to deal with server failure and overloading.

WebCT 0 is a popular and comprehensive web-based course management tool but some functions are too general, e.g., the whiteboard provides paint functions but no structured diagrams. The current version of WebCT does not take account of server failure for the chat and share whiteboard functions, although in an upcoming version WebCT Cobalt 0, it will add load balancing and session fail-over. Dexpo Chat 0 is a web-based Java chat software, which provides virtual conference room with some nice features, such as load balancing and HTTP tunneling. However, the load-balancing is chat room-based, which cannot handle certain situations such as particularly heavily-loaded chat room which overloads any server that it is assigned to. Section 2 discusses the WOTS architecture. Section 3 focuses on the support for collaborative diagramming. Section 4 discusses the implementation and testing of the prototype system.

2. WOTS Architecture

WOTS is a distributed, scalable, reliable wireless online tutoring system, providing textual chat and structured diagramming (UML) support. Users can access WOTS through personal computers or handheld devices.

A message-oriented middleware (MOM) is used for communicating between servers. The MOM can forward messages from the source server to the destination(s). The use of MOM in the system enables the system to be distributed among different computers, even different places, because they are connected to the MOM through the Internet Protocol (IP) address.

The Message Server is the core component in the system, since it handles all user requests. Scalability of the system is achieved by adding additional Message Servers to the system dynamically without restarting the system and interrupting the operation of other Message Servers. Another component manages the cluster of Message Servers, by assigning the most appropriate Message Server to users.

Reliability of the system is improved by fault tolerance and dynamic load balance mechanism. In the fault tolerance mechanism, if any one Message Server should fail, another would take its place and function. Therefore, each Message Server has the same copy of the data to provide redundancy for the system. The dynamic load balance mechanism can distribute the user requests among several servers dynamically when there is an overload server or any additional Message Servers added to system. Hence, the chance of rejecting user requests due to busy servers can be reduced.

The handheld device communicates with the server in the same way as the personal computer, by using Java objects over HTTP connections. Pocket PC is selected as the handheld device to access the tutoring system. Wireless users can use a Pocket PC to access the tutoring system through LAN as well as the Internet.

WOTS consists of five key components (Figure 1): **Message Server**, **Web Switch**, **User Handler**, **Database Handler** and **Java Message Service (JMS) Server**. The

JMS Server is the MOM for the system to handle communications between servers. The client can use web browser or personal digital assistant (PDA) to access WOTS.

2.1. JMS Server

JMS 0 Server is a message-oriented middleware (MOM) for the system. It is used for forwarding messages from one single source to one or more destinations. The JMS Server is capable of handling hundreds of messages every second.

2.2. Message Server

Message Server is used for storing and handling the messages. It stores the messages for each chat room and sends new messages to clients. Each Message Server has a Servlet Engine as an interface layer to accept and handle client requests. The Message Server gets the client messages from JMS Server. In the WOTS, there can be more than one Message Server to provide redundancy and load balancing for the system.

2.3. Web Switch

The Web Switch provides static and dynamic load balancing for the Message Servers. It is also used for keeping tracking the state of each Message Server, such as workload and server failure; and assigning one Message Server to each client based on server workload. The Web Switch has a Servlet Engine as an interface layer to accept and handle client or server requests. It communicates with Message Server through JMS Server and User Handler through socket.

2.4. User Handler

The User Handler is a server for keeping track of the state of each client, such as user weight (the estimation of the number of connections) and login/logout state. It communicates with the Web Switch through a HTTP socket connection as well as the JMS Server to get the up-to-date client states.

2.5. Database Handler

The Database Handler is used to collect chat messages and store them to database. The messages were indexed by their own timestamp. The Database Handler also records the details for each chat room, such as start time, end time and participants. It gets chat messages from JMS Server.

2.6. Hand-held Devices

Users can access WOTS by handheld devices through wireless connection. Since the screen size, system environments and hardware power of handheld device are very different from personal computers, there are two different versions of the client front-end (Figure 2). However, the two versions of client front-end have very similar functionalities. In general, the hand-held client is a restricted version of the PC client.

2.7. Chat Room Management

Tutors can create chat rooms for their teaching subjects (Figure 3). They also can limit the number of student in the chat room and redirect student to another tutor. If there is more than one tutor available for the same subject, the mapping between student and tutor will be based on the number of students in the chat room. The new student will be assigned to the room with least number of students. In addition, the tutors for the same subject can talk with each other secretly.

2.8. Load Balancing

Bourke 0 pointed out that performance metrics are critical because they gauge the workload of the site's implementation. In WOTS, all requests and responses are done over HTTP, hence the most important metric is connections per second, which forms the basis for workload estimation.

<u>Staic Load Balancing:</u>

1. When a client logs in the system, the Web Switch redirects the client to Message Server with the least workload.
2. If a client detects a server fault, it will request another Message Server from the Web Switch. The Web Switch returns the IP of another Message Server based on a static load balancing algorithm.

<u>Dynamic Load Balancing</u>:

1. When the Web Switch determines a server is overloaded (e.g., when it is serving a large number of clients), it will redirect some clients from the overloaded server to other servers.
2. When a new Message Server is added to the system, some clients will be off-loaded from busy Message Servers to the new server.

2.9. Fault Tolerance

Fault tolerance support of the Message Servers adopts the active-active approach. Multiple Message Servers are running, each storing the same set of data providing the same functions. If one of them should fail, other Message

Servers can continue to accept and handle client requests. On the client side, the client detects a server failure by catching exceptions when it is trying to connect to the failed Message Server. On the server side, each Message Server sends periodic live signals to the Web Switch. Hence if the Web Switch has not received the live signal from a Message Server for a certain period, the Message Server is considered to have failed. Clients will not be assigned to this Message Server.

3. Collaborative Diagramming

Concept mapping is a very common technique for knowledge representation in teaching, which has been specialized into many kinds of structured diagrams tailored for individual subjects. In computer science the Unified Modeling Language (UML) 0 is a very popular format for representing the design of information systems. WOTS incorporates an interactive and collaborative diagramming tool to support the sharing and exchange of structured diagrams in the UML format. Similar to the chat room client, the Structured Diagramming clients in the form of clients for a UML whiteboard (Figure 4) communicate with the server through HTTP. Editing events on the diagramming clients create or update graphical objects in the client, these changes are then communicated to the server, which subsequently updates the other collaborating clients.

Because of limitations of screen size, etc. on the hand-held devices, the hand-held client has to be designed differently from the PC client. In particular, screen space is at a premium. Hence graphical objects are not displayed permanently on top of the workspace for easy selection as in the PC client, Figure 4(a). Instead, the hand-held user has to select the graphical objects from a pull down menu when needed, Figure 4(b).

While users collaborate interactively on the structured diagram, conflicts often arise, e.g., when the student and the tutor try to edit the same graphical object at the same time. Locking of objects is supported so that no two clients can edit the same object simultaneously. A color scheme is adopted to indicate which objects have been locked, and by whom (Figure 4(a)).

Drawings on the UML whiteboard can be archived using a combination of XML-based Meta-data Interchange File (XMI) 0 and Scalable Vector Graphics File (SVG) 0. XMI is a de facto standard for storing logical UML models, including the type of objects (use case, class, package, etc.) and the relations among them (generalization/specialization, composition, association, etc.). SVG is used to store the display information, such as the location of the UML objects.

4. Implementation and Testing

The system was developed using Java technologies. Each system server is a Java application. The MOM is a JMS-compliant product OpenJMS 0.7.2. The servers that handle client requests and responses have a Servlet Engine layer based on Tomcat 4.0. MySQL 3.23 is used as the database, connected through JDBC.

Performance of WOTS has been evaluated by simulation. Tests include the following: adding message servers, shutting down of Message Servers, and heavy loading with hundreds of clients. The system performed as expected, without deadlock, exceptions or runtime errors. Further, performance of WOTS was also measured under various loading conditions. It was found that the mean process time per user request as well as CPU utilization decrease as more Message Servers are added, as expected. However, the experimental results showed large random variations when client request turn-around-time was measured from the client side. It appears that the network performance dominates the server processing time and was subjected to unpredictable factors.

5. Conclusion and future work

WOTS is a distributed, scalable, and reliable online tutoring system with static and dynamic load balancing, and fault tolerance mechanisms. The system provides textual chat and structured diagramming support. Tests have shown that the design objectives have been achieved. Work on the next phrase will focus on supporting interactions among students and the tutor in a group discussion environment, and integration of additional supporting tools.

6. Acknowledgement

The work reported in this paper was partially supported by Hong Kong Polytechnic University Research Grant H-ZJ82.

7. References

[1] G. Booch, J. Rumbauch & I. Jacobson. The Unified Modeling Language User Guide, Addison-Wesley, 1999.

[2] T. Bourke, “Server Load Balancing”, O’Reilly, 2001.

[3] Design Expo Network Pvt. Ltd. Dexpo Chat, http://www.dexponet.com/htm2/asp_products/chat.htm

[4] R. Kreutz, S. Kiesow & K. Spitzer, “NetChat: Communication and Collaboration via WWW”, Educational Technology & Society, Vol. 3, 2000.

[5] R. Monson-Haefel & D. A. Chappell, “Java Message Service”, O’Reilly, 2001.

[6] SVG 1.1 Specification http://www.w3g.org

[7] WebCT http://www.webct.com

[8] WebCT, Inc. "WebCT Product Roadmap and Migration Guide", September 2001. http://www.webct.com/service/ViewContent?contentID=5861206

[9] XMI 1.1 Specification http://www.omg.org

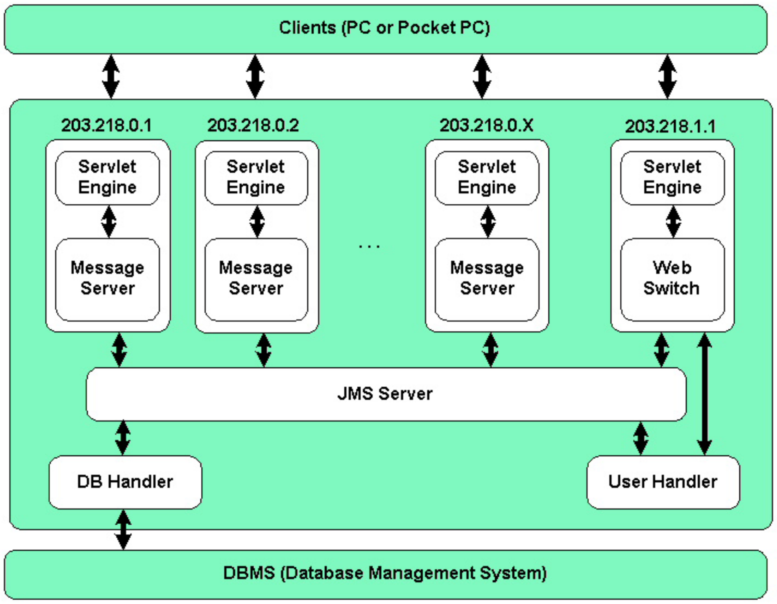
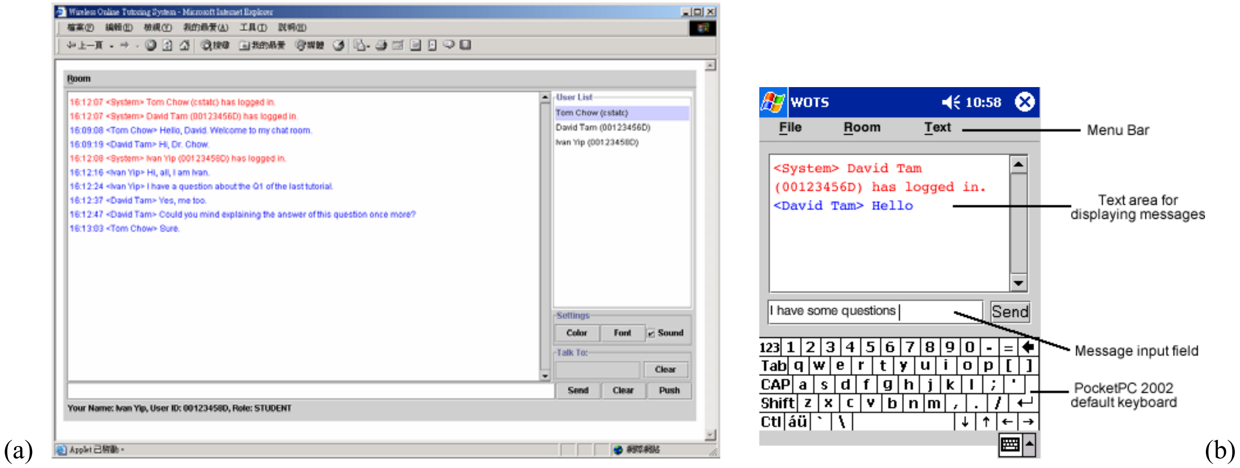


Figure 1. Architecture of WOTS.



16:12:07 <System> Tom Chow (cstatc) has logged in.
16:12:07 <System> David Tam (00123456D) has logged in.
16:09:08 <Tom Chow> Hello, David. Welcome to my chat room.
16:09:19 <David Tam> Hi, Dr. Chow.
16:12:08 <System> Ivan Yip (00123458D) has logged in.
16:12:16 <Ivan Yip> Hi, all, I am Ivan.
16:12:24 <Ivan Yip> I have a question about the Q1 of the last tutorial.
16:12:37 <David Tam> Yes, me too.
16:12:47 <David Tam> Could you mind explaining the answer of this question once more?
16:13:03 <Tom Chow> Sure.

Tom Chow (cstatc)
David Tam (00123456D)
Ivan Yip (00123458D)

Your Name: Ivan Yip, User ID: 00123458D, Role: STUDENT

<System> David Tam (00123456D) has logged in.
<David Tam> Hello

I have some questions

Figure 2. (a) The students' chat room on a PC; the tutor' interface is similar.
(b) The interface of the wireless chat client.

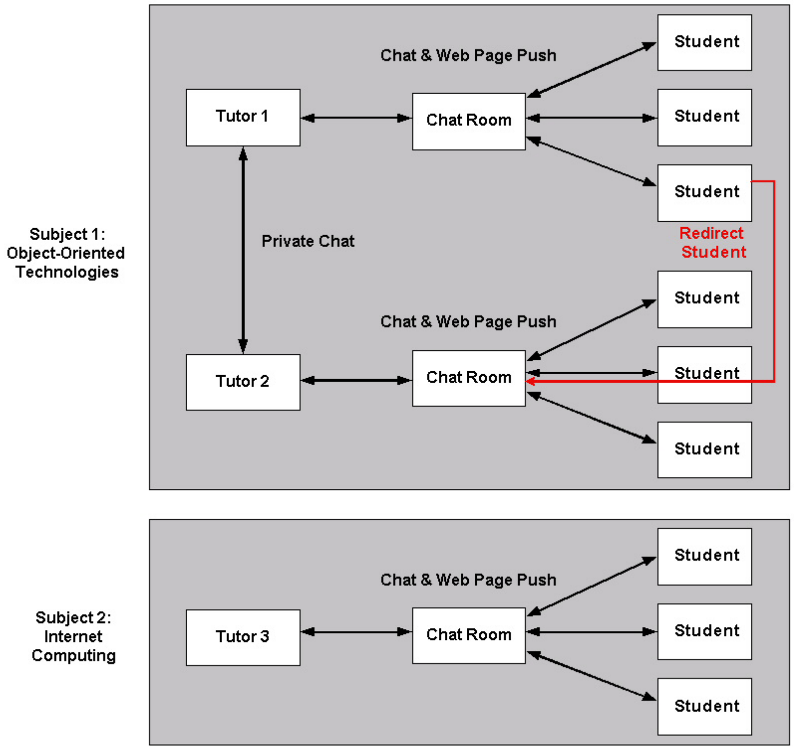


Figure 3. Chat room management in WOTS.

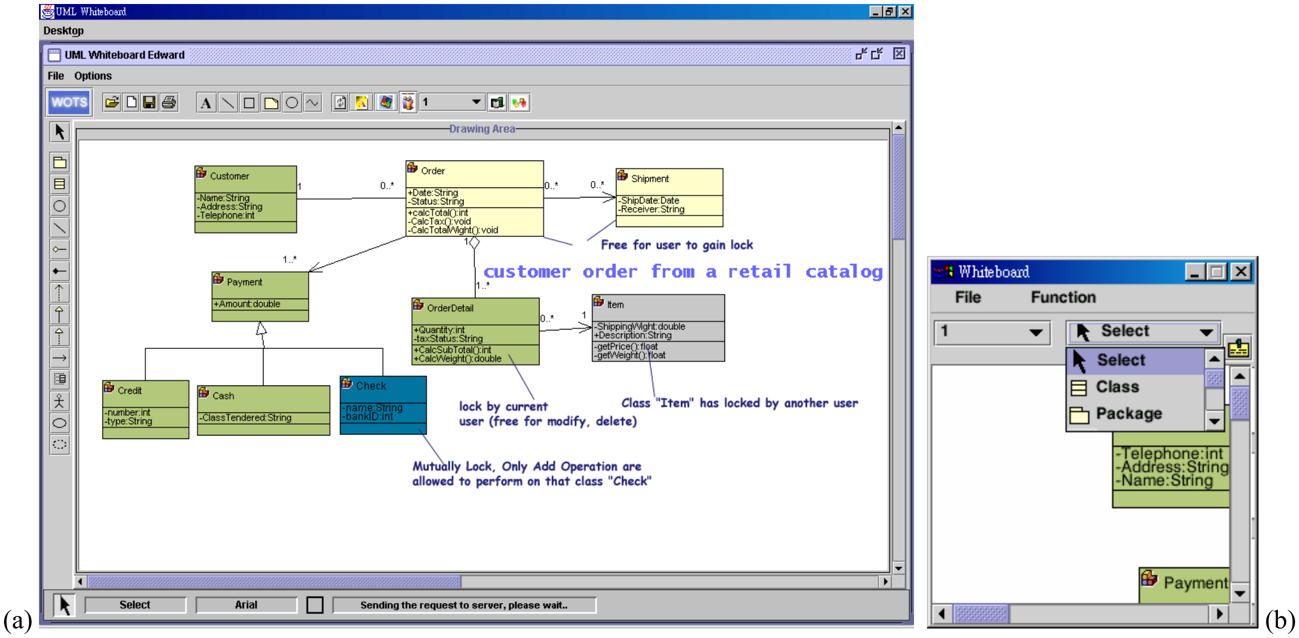


Figure 4. (a) PC client for UML white board with color-coded locks to support collaboration control. (b) Client for Compaq iPAQ.