

UML ARCHITECTURE OF A WEB-BASED INTERACTIVE COURSE TOOL

Lakshmi K. Bhetanabhotla, N. Bussa, A. Mohammad,
O. Abdalla, and Hany H. Ammar

Dept. of Computer Science and Electrical Engineering,
West Virginia University
Morgantown, WV26506-6109
hammar@wvu.edu

Abstract: The demand for educational software is growing exponentially with the surge of interest in educational reform, the Internet and distance education. A Web-based Interactive Course Tool was developed at West Virginia University. The tool provides components for instructors, administrators, and students to interact effectively through the Internet. This paper presents an object-oriented software architecture model of this tool.

Keywords: Distant Teaching and virtual classroom, Internet University, Object-Oriented Analysis and Design using UML, Visual Modeling

1. Introduction

Network - based distributed education is a reality today. This is the result of an increasing demand for distance education for reasons like increased commute time, jammed classrooms and parking lots [1]. It also provides access to quality courses to the learners who are unable to attend classes in the traditional ways. It enables adults and full-time employees to acquire education similar to attending any other school.

Web-based Interactive Course Tool (WICT) is one such instructional delivery environment developed in the Laboratory of Computer-based Instructional Technology (LCIT), at the Dept of Computer Science and Electrical Engineering at West Virginia University [2][3][4][7]. This tool provides support for distance education over the Internet with a high level of interactivity between students and instructors through implementing both Synchronous and Asynchronous modes. The instructors are provided functionalities to add a new course, create course notes, quizzes, assignments, exams and evaluation forms and view their results; participate in the discussion groups and message boards and also post the student grades online. The students are allowed to view the posted course notes and assignments, take the online quizzes and exams, complete and submit the evaluation forms for the instructor, get their grades online and participate in the discussion groups and message boards. The

administrator is granted the privileges to add a new course, a new instructor or a new student to the database, register them for a particular course, query the list of students registered for any particular course, or the list of courses lectured by a particular instructor, etc.,

The focus of this paper is to develop software architecture for this tool and discuss issues related to implementation and testing. Section 2 of the paper presents component-based software architecture. Section 3 discusses the implementation and testing issues, and section 4 provides conclusions and future work.

2. UML Architecture of WICT.

The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. UML provides the application modeling language for [5]:

- Business process modeling with use-cases.
- Class and object modeling.
- Component modeling
- Distribution and deployment modeling.

This project adopted UML modeling techniques for the design. The three kinds of UML models used to describe the component-based model of this project are the Use-Case Diagrams, Component Diagrams and the Class Diagrams. The design begins with the illustration of Use-Case diagrams that represent the system and one or more outside interactors (called *actors*), together with actions performed by the system. Figure 1 illustrates the interaction of the Instructors and the Students with the System. Instructors and Students are the external entities to the System who can log into the System and use the functionality provided by the system. The interaction between the Instructors and the Students can take place in two modes; Synchronous and Asynchronous modes. The Synchronous interactivity is provided by functionality offered by the tools like Course Delivery, Discussion Groups, Feedback tool and Help Session. The course delivery tool and the help session tool are based on the Microsoft Netmeeting conferencing component. The Feedback tool is part of Course delivery and is intended to give the instructor immediate feedback on the impressions of the students during lecture delivery. These impressions include three faces: a smiley face, a doubtful face, and a confused face. Quiz Tool, Grades Tool, Assignments Tool, etc provide support for asynchronous communication. Figure 2 illustrates the interaction of the Administrator with the System. The Administrator is also an external entity to the System who can log into the System and use the functionality provided by the System. The Use-Case Diagrams give an overview of the System functionality and its respective users from a Users' perspective. A more detailed documentation of the design issues from a developer and a maintainer's perspective is the illustration of the Component diagrams. Component diagrams show the software components that make up a reusable piece of software, their interfaces, and their interrelationships. [6]

Figure 3 represents the component diagram of the whole system. The system consists of three sub systems and a Database Component. The three sub systems, Student Sub System, Instructor Sub System and the Administrator Sub System are dependent on the central Database Component.

Figure 4 is the general representation of the components involved in the whole System whose top layer is the Students' and the Instructors' Interface.

The Instructors and the Students enter the System through a Login Tool Component in the first layer, and then the Select Course Tool Component in the second layer. The components in the next layer of the hierarchy are dependent on the session variables that are passed on from the tools in the first two layers. Figure 5 is the representation of the rest of the components involved in the system whose top Layer is the Administrators interface. A more detailed representation of the interfaces that form the first layer in the figures 4 and 5 is represented in the Figure 6.

Figure 6 (a) - represents the class diagram of the Student's Interface in the Students Sub System to the Select Course Tool. Figure 6 (b) - represents the class diagram of the Instructor's Interface in the Instructor Sub System to the Select Course Tool. Figure 6 (c) - represents the class diagram of the Administrator's Interface in the Administrator Sub System to the other components in the Sub System.

Figure 6 (d) - represents the class diagram of the Select Course Tool, which is an interface between the Students and the Instructors Interface to the rest of the components in their respective Sub Systems. A class diagram is a collection of (static) declarative model elements, such as classes, interfaces, and their relationships, connected as a graph to each other and to their contents. A detailed class diagram representation of each of the components represented in the above diagrams is documented in [7]

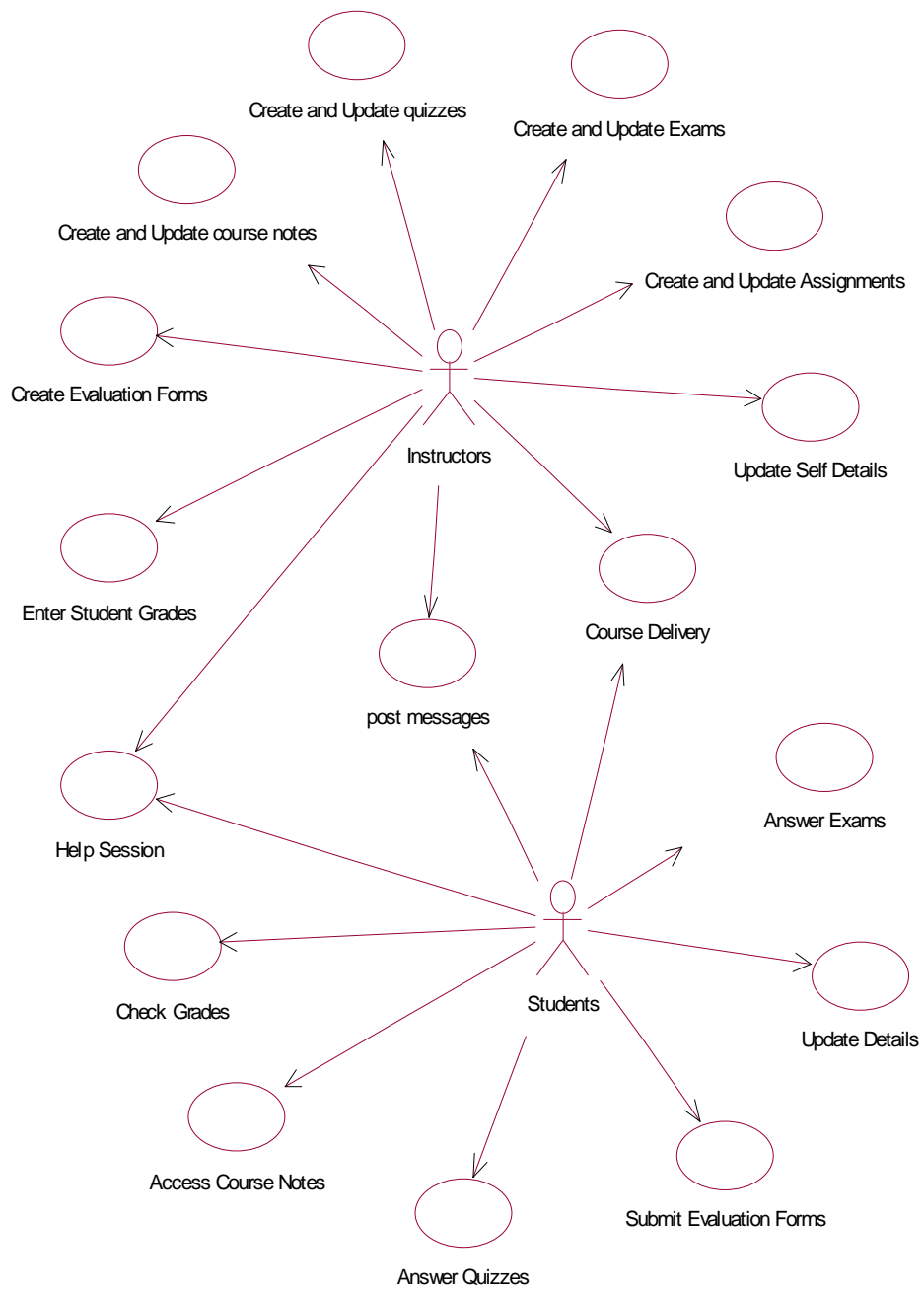


Figure 1 – Use Case Diagram with Instructors and Students as Actors.

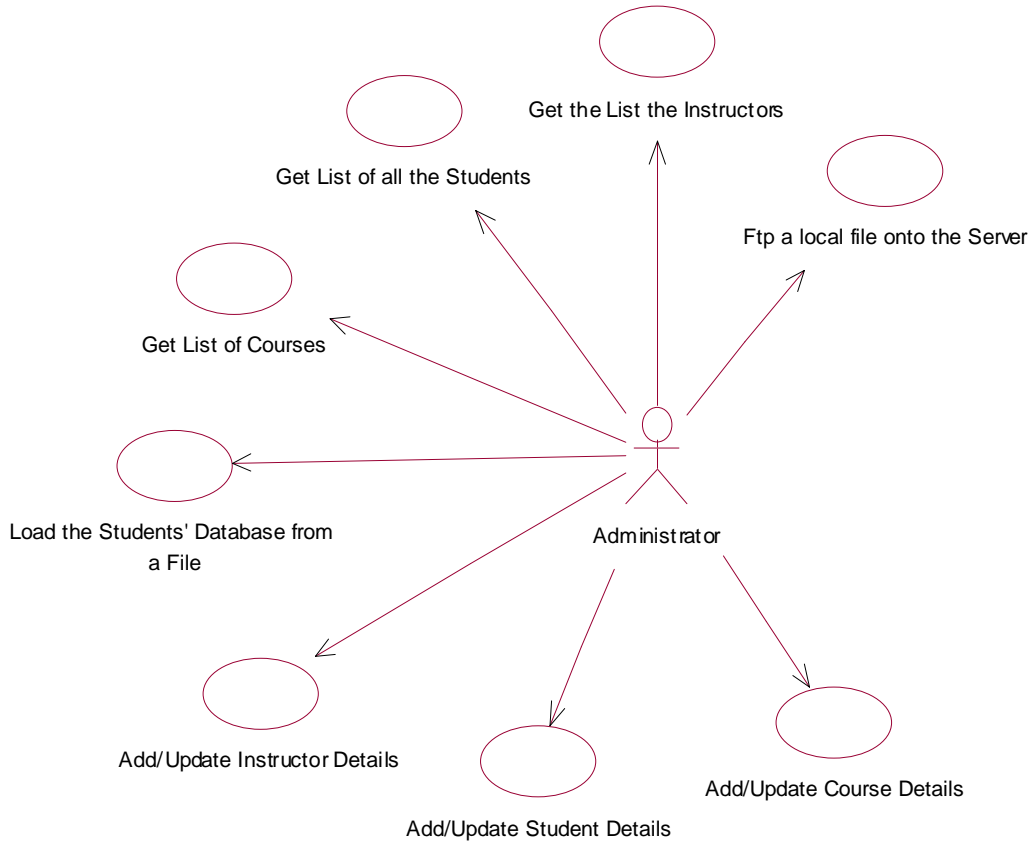


Figure 2 – Use Case Diagram with Administrator as the Actor

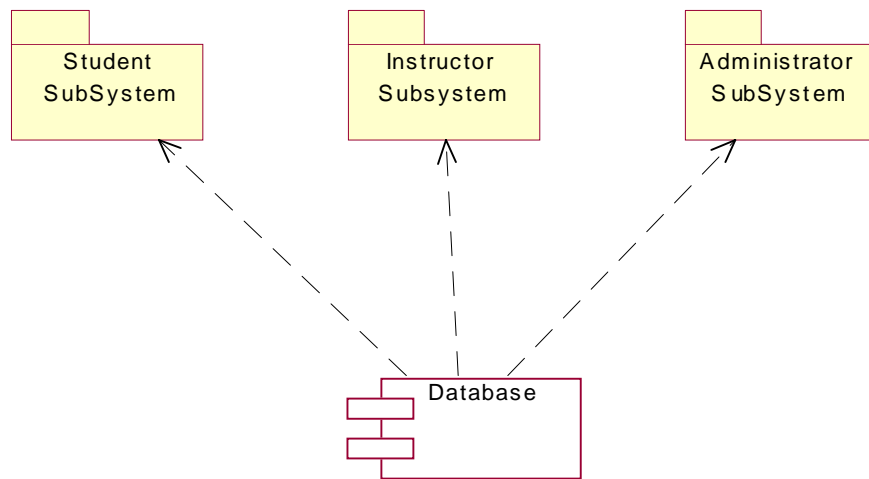


Figure 3 - Component Diagram of the whole system.

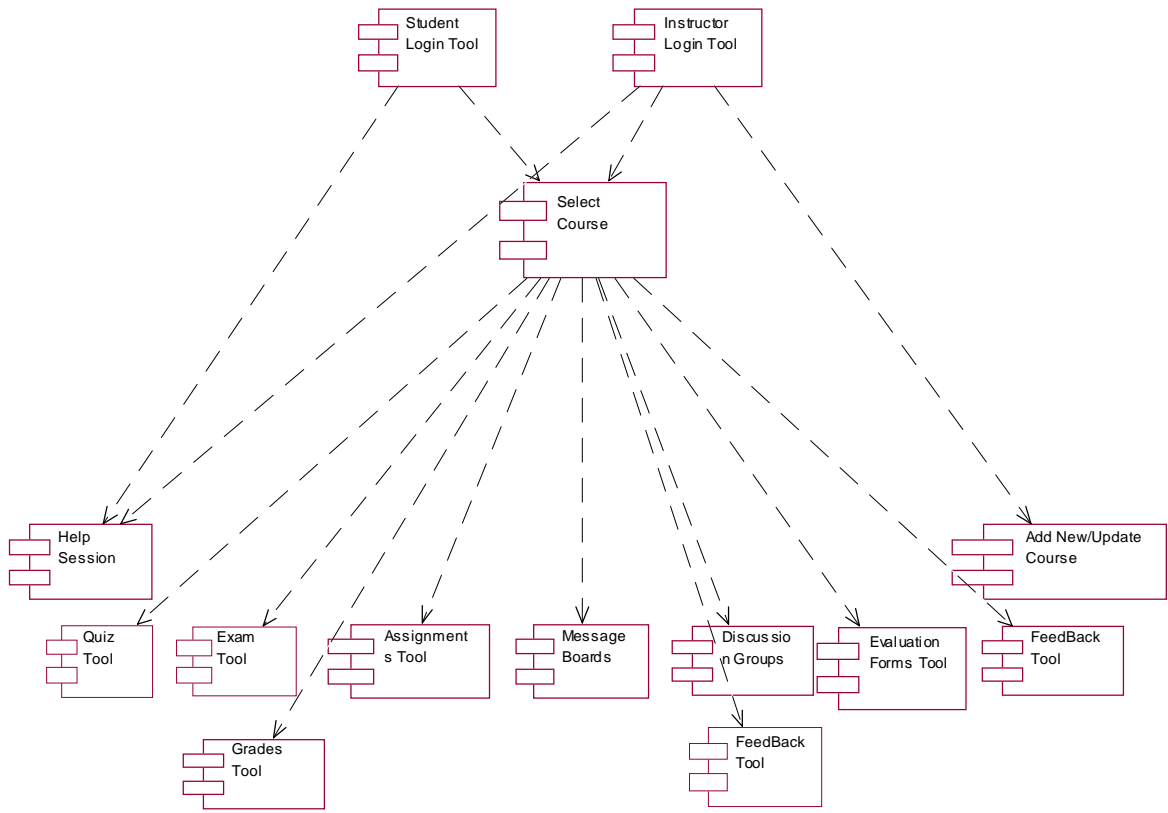


Figure 4 – Component Diagram for the Instructor and Student Sub Systems

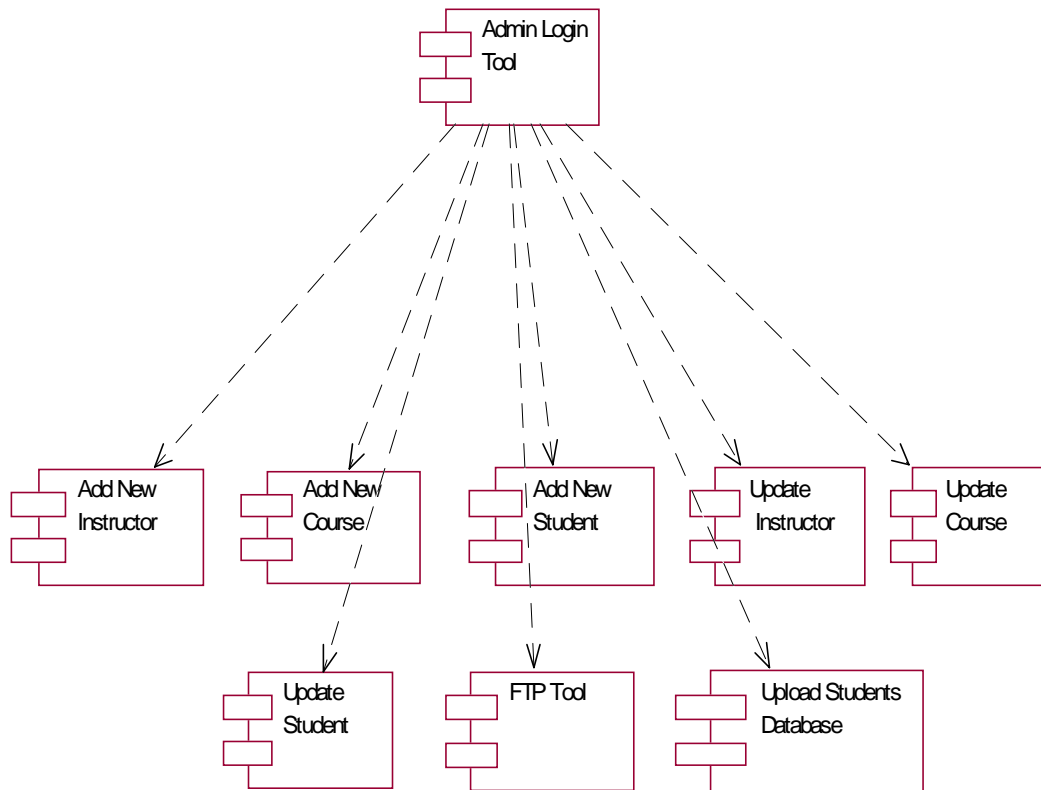


Figure 5 – Component Diagram with Administrator’s Sub System.

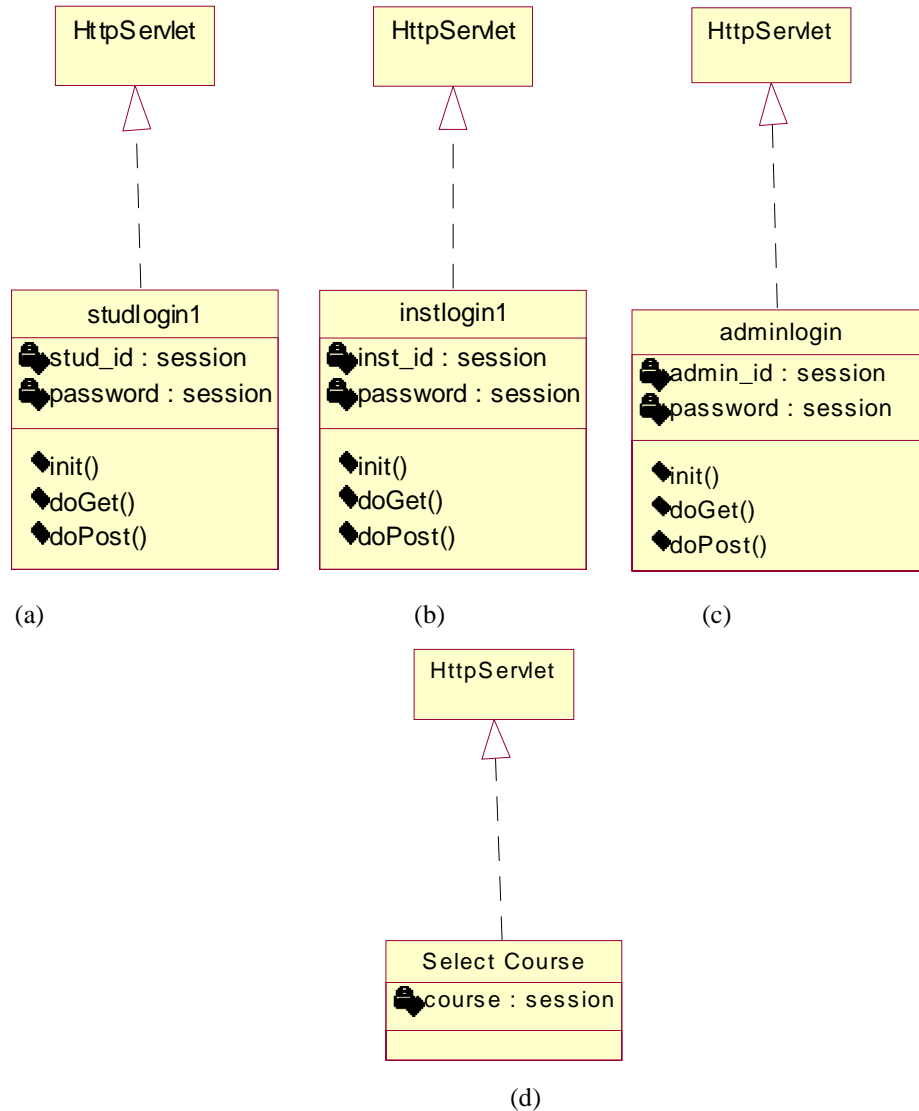


Figure 6 – Class Diagrams of the Interfaces

3. WICT Implementation, Deployment and Testing

The above architectural design leads to the implementation of the Tool as a three tier Client Server model. The first tier is the Web Clients who log on to the Tool through the Internet; the second tier is the Application Server/Web Server and the third tier being the Database Server. The WICT prototype was in fact implemented as a two-tier Client Server architecture where the Database Server also resides in the second tier. The advantage of providing the 3-tier flexibility is the ability to deploy the Database Server on a

different machine to support scalability. JavaWebServer2.0 was used as the Application/Web Server and the IDSServer was used as the Database Server in implementing WICT. The tool can successfully be deployed on all the machines running Windows 3.x/95/98/NT and above. The components in the Tool were implemented using the following technologies: Java Swing API, Servlet API, JDBC API and Java Networking APIs.

The testing of Web-based applications has some similarity with the testing of desktop systems, which involve the need to test the usual functionality, configuration, and compatibility, as well as performing all the standard test types. However, Web application testing is more difficult because complexities are multiplied by all the commercial components that interact with the application. When an error is located in a Web environment, it's often difficult to pinpoint where the error occurs. The behavior seen or the error message received may be the result of errors happening on different distributed components of the system and the error may be difficult to reproduce. Web environments are dense with error-prone technology variables. The five fundamental considerations of Web-application testing are:

- When an error is flagged on the client side, it may be the symptom of an error instead of the error itself.
- Errors may be environment-dependent and may not appear in different environments.
- Errors may be in the code or in the configuration.
- Errors may reside in any of several layers.
- Examining the two classes of operating environments - static versus dynamic - demands different approaches.

The WICT prototype has been tested in the Lab and the successful and failed test cases are documented in [7]. Testing of the tool components on actual courses is yet to be conducted.

4. Conclusions

A UML software architecture model of a Web-based Interactive Course Tool is proposed. The architecture contains components to support interactions between instructors and students. These components include a help session components and a course delivery components base on NetMeeting, the Microsoft conferencing tool. A simple prototype of the tool is implemented, deployed and then tested. The scalability of the proposed architecture can be enhanced using a distributed component architecture based on a broker framework such as CORBA (Common Object Request Broker Architecture). The broker services could be utilized to develop an extensible architecture. The use of a conferencing tool component for student instructor interactions could pose

performance problems in terms of unacceptable network delays and degradation in the quality of audio-video interactions.

5. References

- [1] Mark, J Pullen. "The Internet-Based Lecture: Converging Teaching and Technology", ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE), July 2000.
- [2] NagaRaju, Bussa. "WVU - Interactive Web Based Distance Learning Tool", MSEE Thesis, Department of Computer Science and Electrical Engineering, West Virginia University, December 1999.
(http://etd.wvu.edu/ETDS/E1116/bussa_n_etd.pdf)
- [3] Osama S. Abdalla, "DBAdmin : A Web-based Application for Distance Education", A Masters of Computer Engineering Problem Report, Dept. of Computer Science and Electrical Engineering, West Virginia University, December 1998.
- [4] AbdulMobeen Mohammad, "WVUCT: A Process Centered Environment for Distance Education", MSEE Thesis, Dept. of Computer Science and Electrical Engineering, West Virginia University, March 1998.
- [5] The Unified Modeling Language, <http://www.rational.com/uml/gstart/index.jsp>.
- [6] Gomaa, Hassan "Designing Concurrent Distributed, and Real-time Applications with UML, Object Technology Series, Addison Wesley, 2000, ISBN 0-201-65793-7.
- [7] Lakshmi Bhetanabhotla, "Enhancements of WICT", A Masters in Computer Science Problem Report, Dept. of Computer Science and Electrical Engineering, West Virginia University, October 2000.