Supplement to Distance Learning: Design for a Remote TA Support System

Hangjin Zhang, Kevin C. Almeroth
Department of Computer Science
University of California
Santa Barbara, CA 93106-5110
{hangjin, almeroth}@cs.ucsb.edu

Abstract: Usually one or more Teaching Assistants (TAs) are working onsite to help students during discussion sections, during office hours, or with lab assignments. However, in an environment where distance learning is used, an onsite TA might not be available. In this paper we propose a Remote TA Support System to provide anywhere-and-anytime TA assistance. The system provides both the students and the TA with a rich set of interaction options: either synchronous chatting or asynchronous messaging; and either private one-on-one meetings or a many-to-many group discussion. The collaboration methods between students and TAs include text chatting, image annotation, screen/application sharing, presentation and bi-directional audio/video communication. The system also includes an archive server that provides content-on-demand for later reference and evaluation. In this paper we identify the features and requirements of such a system and plan a set of phases to develop the system.

Introduction

With the rapid development of computer networks, people can access information and communicate with others without being constrained by space and time. In the field of university education, in addition to conventional correspondence courses and broadcast-type courses, various forms of distance learning using the Internet have become possible. Now students are not limited to obtaining knowledge only within the physical bounds of a classroom. Digital classroom and distance learning projects enable students to access lecture or seminar videos in an on-demand and from-anywhere fashion. However, one key learning method, the use of teaching assistants, is still restricted by time and location. Most distance learning projects seem to neglect the role of teaching assistants, even though they play an important role in learning.

To seek assistances from a remote teaching assistant (TA), a commonly used method today is email, or less commonly by telephone. These methods are not closely integrated into most current distance learning systems, and both have drawbacks. In particular, email is not a real-time communication and is unlikely to obtain a fast response. In addition, most email systems today do not support transaction and workflow processing. Lotus Notes is an example application that supports these functions, but it is too heavyweight for most classroom uses. The telephone is connection-oriented and real-time, but it only provides aural information. The lack of visual information in telephone communication makes it hard for a student to clearly describe the question/problem. A better solution might be instant messaging (IM). It combines some of the advantages of both email and the telephone, but creates its own disadvantages, for example, its inability to exchange lengthy text representing complex ideas. Furthermore, in a practical setting, students and TAs are unlikely to exchange IM account information.

Facing these challenges, our goal is to first design, and then implement, a Remote TA Support system. The goal of the system is to enable students to learn anywhere and anytime. This goal does not create the requirement of learning *all* of the time. Instead, it means whenever the user (student) is seeking assistance, the corresponding information will be available, regardless of the location. Besides availability, we also focus on the challenge of remote interaction and remote collaboration between the two types of users. We believe successful learning is student-centric, full of interaction, collaboration and feedback. The proposed Remote TA system is not a platform meant for communication only, instead, it is a communication platform aimed at interaction and collaboration. This paper tries to identify the unique problems of remote collaboration arising in a TA setting.

By analyzing the real world teaching assistance scenarios, we create a list of system services accordingly. We then translate these scenarios into specific application services. Also, we design a high level architecture to accommodate required services, and establish a set of phases to eventually build a fully functional system.

The remainder of the paper is organized as follows. In the next section, we analyze several scenarios in which teaching assistance is used. These requirements are then used to create a set of Remote TA Support System objectives. Next, we map these real world scenarios to application services and specific functions. Then we describe our system design and the set of phases for implementing the system. Related work is described, and finally, the paper is concluded.

Conventional Teaching Assistance Settings

In order to better understand the design principles of the Remote TA system, we first examine the conventional teaching assistance scenarios and communication types.

Scenarios of Teaching Assistance

In a conventional learning environment, there are three situations in which a student may need to meet with a TA: a discussion section, office hours, and an experimental- or lab-oriented section. The third scenario is further divided into two sub-categories: computer programming labs and physical experiments.

In the discussion section scenario, many students and a TA meet together in a classroom. The TA plays a role similar to that of an instructor. It differs from a common lecture in that it involves more interaction between the students and the TA. The number of students varies from only two or three to 30 or more. The essence of a discussion section is a public "broadcast" session: the TA broadcasts information to all students present, and any question raised by a student is heard by all. Students can join or leave the discussion section freely. It differs from a typical presentation though. In a normal presentation, the topic is determined by the presenter, and the information flow is more or less one way: from the presenter to the audience members. The only interaction in a presentation is feedback such as audience questions. In a TA discussion section, however, the set of topics is more ad hoc, i.e. usually significantly influenced by frequent questions from the students. The TA may frequently change topics based on students' interests. Discussion sections, if done well, involve much more student interaction.

The office hour scenario is quite different from a discussion section. The TA is waiting for students in a specified office during a specified range of time. The conversation is usually private, i.e. only one or a small group of students are involved. The office hour scenario is also question-centric. In a question-based conversation, students may bring textbooks, homework, quizzes, or code to the TA as a shared reference. With the help of this common reference, it is usually easier to solve a question or problem. This question-centric and context-sharing nature distinguishes remote teaching assistance from many other types of remote interaction.

The experiment scenario has properties of both of the two other scenarios. Typically, all students conduct an experiment in a specific laboratory either individually or in a group. The TA is also in the same laboratory to monitor progress and provide "over-the-shoulder" help. A student can seek personal assistance from this on-site TA. The TA can also announce (broadcast) a direction or instruction to the whole class and monitor resulting action. An important task the TA performs is to detect "abnormity". When a student or group does the experiment incorrectly, the TA is responsible for seeing this and providing additional guidance. The guidance can be done either by physically manipulating the student's instruments or by showing a demo with another set of instruments.

All of these three scenarios require the TA being present in the same location at the same time with the student(s). Interaction and collaboration between TAs and students can be achieved naturally because they can talk face-to-face.

Types of Communication

In conventional learning situations, there are a variety of mechanisms for communicating. Not only is there audio and visual communication but a number of additional methods for communicating shared information like books, notes, or computer programs. The goal then is to classify the methods by which this communication can be shared among remote participants. This allows the context to be preserved between the two participants. The following is a list of communication types that together attempt to preserve the whole spectrum of information that exists in an interactive session:

- **Telephone**: provides good real-time voice communication. It can also support multi-participant conferencing. However, unless the service is provided using Voice over IP (VoIP), the telephone is not integrated as part of the Internet.
- **Short Messaging System (SMS)/Multimedia Messaging System (MMS):** these are value-added services in modern 2G/3G wireless phone system. They provide non-voice information over the public phone system. Like the telephone, these kinds of services might be used in the case where users are not connected directly to the Internet.

- **Email**: even though it is not real-time, email is quickly becoming a primary means of communicating between students and teaching assistants and instructors.
- **Instant Messenger (IM)**: while IM provides better promptness and online status notification, it is not commonly used in an instructional environment.
- **Video Conferencing**: not only being able to hear voice from a remote participant, but also the video enables users to view the environment of the remote participants. The challenge with audio and video is having the facilities at both ends and the network capability to be able to transmit and receive a signal of acceptable quality. For this reason, even though video conferencing is often cited as a good solution for remote interaction, it is only effective in a limited number of environments.
- **Shared White Board**: a shared area is created for two or more participants for drawing. Every participant gets the same white board status and result at any given time. There are a number of good systems available. The challenge is integrating it into a larger system.
- **Object Sharing**: for some electronic objects, such as text blocks, images, it is possible to be share within multiple distributed applications. An effective object sharing system is important for sharing information, and yet it is often overlooked in collaborative systems.
- **Screen Sharing**: a more specific and slightly different kind of communication than object sharing, screening sharing allows users to see what is on each other's screen and also possibility to control it.
- **Application Sharing**: this is an extension to screen sharing. Besides viewing the remote computer screen, it is also able to control the remote computer resources using a local GUI interface.

Remote TA System Challenges

In a distance learning environment, because the TA is physically remote from the students, the TA cannot observe them during learning sessions in the same way as during a face-to-face meeting. As a result, many challenges are created. Some challenges are common to all remote collaboration applications, while some are unique to the remote TA system. In this section we list the major challenges that must be solved.

First, like with other remote interaction and collaboration applications, the TA needs information about the students and their activities. This information needs to be provided contextually, e.g. a copy of the student's computer code or the page in the textbook that has caused some confusion. This requirement holds for all three scenarios described in the previous section.

Second, the TA may also need the ability to use or control one or more remote physical devices, especially for a network connected device such as a computer. Like an in-person session, the TA needs to be able to turn to the right page in the book, bring up on the screen the right part of a computer program, or enter some information into an application. Being able to actually do this remotely is much more effective than instructing the remote student on what to type or what information to access.

Third, even if some communication mechanisms are already available, we need an integrated platform to take full advantages of all of the possibilities. Otherwise, using a heterogeneous group of communication tools may cause consistency and efficiency problems. In addition, it is also a challenge to get different information from one application to another if they are not tightly integrated. For example, integrating voice and video to achieve "lip-sync" would be very difficult if audio and video were sent with different tools.

Fourth, some successful solutions for other remote collaboration systems may not be perfectly suitable for a Remote TA System without substantial changes. Most existing systems are not designed purposely for the three scenarios described previously. Video conferencing, for example, is less helpful in a remote TA situation than for a distributed meeting. The information flow in a presentation is mainly one-way, from the presenter to audience members. Although audience feedback or questions also occur in a presentation, it is not a key part of the presentation and is very rarely handled well, especially when those asking the questions are themselves remote.

Finally, there is a challenge inherent in the often asymmetry between students and the TA. For many remote collaboration systems such as [2] [6], their goal is to enable distributed partners to work on a common object to accomplish a shared task. Therefore, they focus on issues such as shared resource access, synchronization, and version control. In a remote TA environment, however, the TA and students do not have equal roles, and they are not collaborating to accomplish a shared task. Instead, the important issue in remote assistance (especially in an experiment scenario) is to enable the TA observe a student's activities, and give useful guidance.

Services of the Remote TA System

The Remote TA System is not a platform meant for communication only, instead, it is a communication platform aimed at interaction and collaboration. The main technique used to improve interaction and collaboration is to share *context* between either individual students or a group of students and the TA. We consider the communication functions and collaboration functions as different layers of a stack. This representation is shown in Figure 1. In this figure, some of the important services are also listed for the different layers.

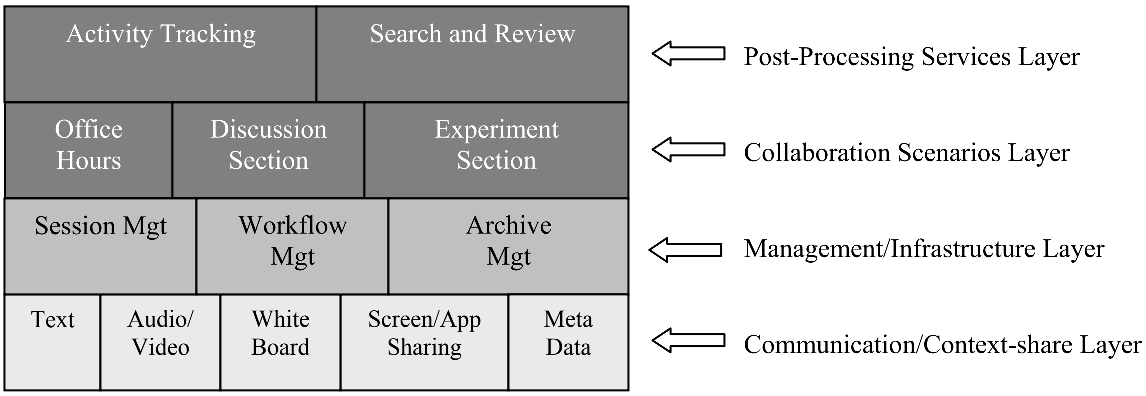

Figure 1. Layers and Services in the Remote TA System

The bottom row of Figure 1 includes all of the basic interaction methods. Each provides different parts to the problem of context sharing. Exchanging text, audio/video, and whiteboard are straightforward communication methods for which there is a good deal of experience. Screen and application sharing are emerging communication mechanisms and important for effective remote interaction. Through the various communication methods, one user can show the screen or demonstrate an application to another. The second user can provide feedback on whether he/she understands. Metadata provides additional support for audio/video context sharing. By embedding metadata into a video, a user is presented with more information than just the audio and video. For example, using "index" metadata for a session, a user can quickly navigate to the desired position of a video clip. Furthermore, subtle information that is hard to make visible in a regular-sized video can be presented as metadata and accessed when needed.

The second layer in the figure is the infrastructure layer. The interaction and collaboration between students and TAs is known as a session. A session can be between two or more participants. Resources and context are shared within a session. It manages user joins and leaves, keeps active session status, and allows a TA to have a public session plus additional private sessions simultaneously.

The Remote TA System is designed to provide several workflow functions. These functions are provided by the workflow management module. When two or more TAs are available for a distance learning course, they cooperate using workflow management. For example, if a student asks a question by posting a message, and one of the TAs answers the question, the system should mark the question as being answered so another TA does not also answer the question. This module also provides a mechanism to automatically generate reminders or suggestions to students based on their workflow status. For example, students may be given "suggestions" on which modules to review if they miss a session or do not do well on an assignment.

All session history is recorded in the server by the archive management module. We try to store as much of the interaction as possible in the archive. There are two levels of information in an archived document. One level is all of the data generated by session participants, along with synchronizing information such as timestamps. Through this level of data it is possible to re-play a session. The other level of data is optional metadata that helps viewers to better navigate and retrieve documents. Examples of such data are session summaries and embedded hyper-links of objects that were used in the session.

The next two rows in the figure represent the three scenarios and some post-processing services. In the next two sections, we describe how these scenarios translate into specific services that will be implemented in the Remote TA System, and then introduce two post-processing services aimed at enabling asynchronous learning, which is otherwise difficult to achieve in a conventional environment without computer and networking support.

Translating Between Scenarios and Services

Our goal is to reproduce, as much as is possible, the traditional TA environment. We hope to accomplish this goal by supplying computer network support for remote human interaction and remote instruction. This section describes how to translate three conventional TA scenarios into system services.

Discuss Section Scenario: In this scenario, the main function is to convey the TA's lecture to all students. With the help of the various interaction methods, the TA and students can be in separate physical locations. To realize the interaction goal, the feedback function is crucial, especially on the student side. Some previous work [5] provides good feedback mechanisms for distance learning scenarios. Because of the similarity between a normal lecture and the Discussion Section scenario, the same feedback mechanisms can also be applied in the Remote TA System. In this scenario, the session is initiated by the TA, and the properties of the session are set to public. In this way, all students can join the session. The TA can choose to broadcast a lecture to all students with a significant amount of supporting material, e.g. text, audio, video, slides, or shared workspaces.

Office Hours Scenario: In the real world, because the TA and the student meet together, they can refer to a common set of objects, e.g. papers, books, a computer program, or content on one of the participant's screen. This material helps facilitate communication. The remote TA system needs to simulate the context sharing that occurs in the real world. The type of material that forms the base for the shared context can vary. It could be a common web page, a shared whiteboard, a screen snapshot of the student computer, or even the student's or TA's computer screen. During the conversation, a TA may also need to manipulate one of a student's devices, e.g. a student's computer.

Experiment Scenario: In this scenario, a TA should be able to assist a remote student in completing an experiment. The collaboration requirement in this case is much harder and requires a much more complicated set of tasks to be completed. As described above, the TA should be able to monitor all students and what they are doing. We divide experiments into two categories: computer programming experiments and physical experiments. The difference between the two categories is how to display and manipulate the experiment. The former is done via a computer screen, while in the latter category, the experiment is not shown on a computer screen but rather through video or some other source. Examples in the second category include physics, chemistry, and electronics experiments. For computer programming, because the results are shown on a screen, the TA can easily catch the working environment or context by viewing the student's screen. Quite a bit of software has been developed to do screen sharing. In addition, some of this software also offers remote control functionality, e.g. Microsoft Remote Assistance [9]. For this second type of experiment, in order for the TA to see the working environment or context from the student side, some conversion must first be done. A typical way is to use live video. The student side uses a camera aimed at the experiment, and transmits live video to the TA. By watching the live video, the remote TA can also monitor what the student is doing. If the TA needs more information, the metadata mechanism can also be used. For example, results data can be fed into the student's computer and passed to the TA in synchronization with the video.

Post-Processing Services

Since all of the remote TA activities are recorded by the archive server, it is possible to provide some value-added services based on the recorded data. One intuitive service is search and review. A student can easily view a missed discussion or office hour session at a later time. Because the archive module records synchronized conversations within a section, a student may not lose much contextual information during a later playback. As part of the Remote TA System we will focus on how to facilitate effective searching by adding proper index and metadata to the archived session data [11].

A more subtle and valuable service is student activity tracking or monitoring. Some sample activity information includes:

- Who attended the session (office hours, discussion sections, or experiment sections)?
- Who reviewed an archived session record?
- Who asked questions during a section?
- What are the questions?
- What is rating of the questions in terms of generality? (i.e., is the question also useful for others?)
- What is the context for the question?
- What are external resources related to the question?

By collecting and analyzing this information, usage patterns for each individual can be identified. These usage patterns can be used to evaluate the effectiveness of the remote TA system and to provide adaptive individual reminders or suggestions to students. For example, when the system finds a frequently asked question, either asked during office hours or accessed via the web page, all students who have not asked the question or searched for the answer can be told that this is a confusing concept and that they should review the material. Furthermore, students who do not review material from discussion sections or office hours can be sent reminders that only by studying supplementary material can the student expect to do well.

Architecture and Development Phases

Having discussed the scenarios and functions of the Remote TA System, we now describe its architecture and the phases in which an actual system could be developed. From a high level view, the remote TA system contains three components: a StudentClient, a TAClient, and a RemoteTA Server. Both the students and TAs login to the RemoteTA Server using their own client. The conceptual connection model is illustrated in Figure 2.

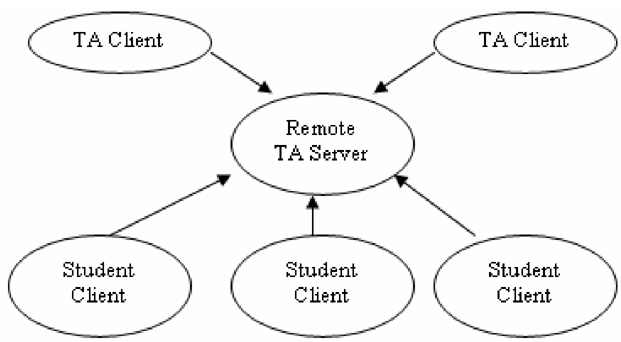

Figure 2. Connection Model

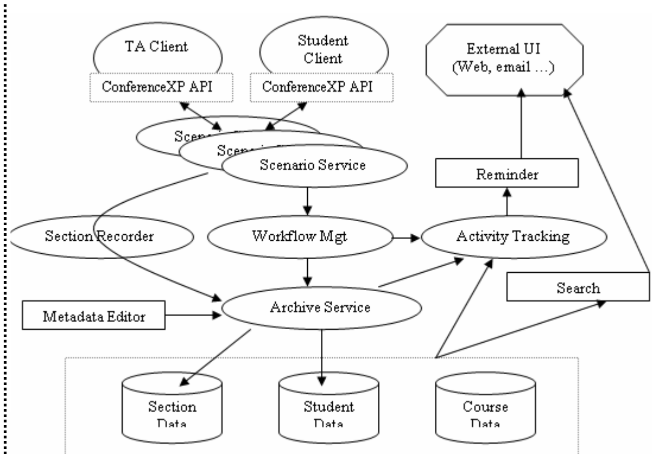

Figure 3. Architecture

The two clients provide GUI interfaces to users, one each for the students and TA. The central server manages active sessions and archives them. Note that this connection model does not imply that the central server acts as a message relay necessarily, because in an environment where multicast is enabled [12], any client can communicate with other interested clients as well as the central server. There is no need to pass messages to the central server first.

Our focus in this paper has been to develop a design for a workable Remote TA System. Figure 3 illustrates the overall architecture. Clearly the system still works even if some modules are absent or not fully developed. Therefore, we divide the development process into multiple phases, with each phase having its sub-goals and deliverable functions. The rationale we used for developing a phased approach is based on the following:

1. **From simple to complex**. Having a basic, functioning system is much easier to test and forms a better base for developing and testing advanced functions.
2. **Feasible and useable at each phase**. A coarse yet workable tool is better than a delicate shell without content. At the end of each phase, we expect to have a usable version.
3. **Minimal initial equipment requirements**. As more functionality is added, more specialized hardware is needed. The ability to use the Remote TA System in almost any situation is critical to its usability. Only after we have had measurable success with the basic components can we justify building a more complex system. Furthermore, if a high level of effectiveness can be achieved without specialized hardware, all the better.

Phase 0: Basic Infrastructure, Session Management

This step includes the construction of the core of the RemoteTA server, and the design of the interface for the two clients. The most important session management functions are implemented in this phase. A basic version of archive module is desired. Message format and communication protocol are also defined. For the interface design issues, there is a tradeoff to be made between choosing a web-based and an application-specific interface. Users are already familiar with web interfaces, and the only client tool needed is a browser. But the web-based application has some functional limitations, for example, it is hard to take advantage of some off-the-shelf toolkits, such as the Microsoft ConferenceXP Platform [10].

Phase 1: Text Chatting

In this phase, we plan to develop an instant messaging interface that supports text chatting. Basically this phase is to verify that the session management module developed in Phase 0 works. We will also build the client framework for latter phases. In this phase, almost no contextual information can be passed. Unlike popular instant messaging services, our text chatting tool will support more modes. A student can initiate a new chatting session with an available (online and not busy in other sessions) TA, or can join an existing public chat session. A TA can also initiate a chat session. The chat messages are archived for later retrieval.

Phase 2: Image, Slides and Annotation

After Phase 1 is complete, Phase 2 will add a shared whiteboard to the chat service. In this whiteboard, all of the session participants can draw pictures that are transmitted to other participants in real-time. This function can also be used as a rudimentary screen sharing service. A user can copy and paste his/her screen snapshot to the shared board. The TA can also playback presentation slides using the whiteboard.

Phase 3: Screen/Application Sharing and Remote Control

In this phase, we plan to implement a true screen sharing mechanism, i.e. the student's screen output is transmitted to the TA's computer in real-time. The TA can also control the student's computer remotely. This function is useful in helping with real-time debugging of programming assignments. For example, the TA can help debug the code in student's environment.

Phase 4: Bi-Direction Audio/Video and Metadata

The implementation of previous phases does not require any specialized hardware, e.g. a camera or microphone. But for non-computer-based experiments, there is other information that cannot be communicated as an object or shared on the screen. An audio/video conversation is required. The TA is able to receive information about the experiment by watching and listening to the audio/video feed. With this functionality we can also support straightforward video conferencing.

Phase 5 Archive Searching, Workflow Management, and Post-Processing Services

These functions are all enhancements to the RemoteTA server. If a student needs help when the TA is not online, the system searches the archived material for related topics and generates suggestions that may be of assistance. This is our attempt to provide a "Virtual TA" system, i.e. TA support when no human is actually available to give help. Of course, this step also requires a good system for archiving, searching, and accessing all of the material and context of prior sessions with the TA.

Related Work

For each of the teaching scenarios, there have been several related projects. Because of the similarity between the Discussion Section scenario and a lecture, many of the distance learning techniques are applicable. However, most of this work lacks real focus on the importance of interaction. For the Office Hours scenario, numerous commercial Instant Messaging systems provide limited functions for remote assistance.

Also, there has been some work in "remote experimentation" [1][7][8]. The most important feature of the existing work is that the students remotely configure experimental equipments and receive real-time video/audio feedback. Students do not physically touch the equipment. They operate equipment through a web interface, and check the result of the operation using a real-time video feed. Our Remote TA System described in this paper works using the opposite approach: students physically operate real equipments, and TAs get a real-time view of the experiment.

Remote Assistance is a technology in Windows XP that enables Windows XP users to help each other over the Internet [9]. With this tool, one user, called the "Expert", can view the desktop of another user, called the "Novice". With the Novice's permission, the Expert can even share control of the Novice's computer to resolve issues remotely. In our Remote TA System, we introduce a similar function to enable a TA to help students with programming assignments.

Some research has been done to construct a "virtual TA" session with a student [1][4]. The automatic reminder and suggestion mechanism in our system partially represents this functionality. In this case there is no *real* TA, just a virtual one. This system seems to be insufficient because we believe that human reactions are very complex. It is difficult to simulate human behaviors by computers. Therefore, unless the setting was very straightforward, it would be more desirable to have a human TA than a computerized TA.

The Open Remote Collaboration Tool (OpenRCT) [3] is a multidisciplinary effort to enhance collaboration, collaboration between students working together, between students and instructional staff, and between researchers who are not co-located. OpenRCT is now mostly used as a second-language learning tool in distributed campuses. It shares a lot in common with the Remote TA System. Still, there are significant differences: the audio/video transmission in OpenRCT is not real-time. Instead, it is transferred by downloading a file. Furthermore, OpenRCT does not aim to support the experiment scenario.

Conclusions

We have presented in this paper the design of a Remote TA Support System, which has merits both for the students and TAs. Remote TA topics have not been widely researched in distance learning. There are unique challenges that must be overcome. The most important of which is that in a remote TA setting there is interaction that is based on collaboration and context. Most existing communication tools do not meet both requirements. Furthermore, many existing systems, such as email and instant messaging, are based on the assumption of symmetric interaction. The Remote TA Support System introduces a full set of interaction and collaboration options, with TAs and students playing different roles in the system. As part of our design, we have also planned the development phases for implementing the whole system. The actual implementation and evaluation are left for future work.

References

[1] H. Mitsui and H. Koizumi. "A Remote Experiment System Provided with Individual Guidance Function and Report Preparation Support Function", 17th International Conference on Advanced Information Networking and Applications (AINA'03), Xi'an, China, pp. 117-124, March 2003.

[2] H. Gajewska, J. Kistler, M. Manasse, and D. Redell. "Argo: A System for Distributed Collaboration", ACM Multimedia, San Francisco, California, pp. 433-440, October 1994.

[3] OpenRCT: http://www.opoenrct.org/, December 2003.

[4] G.-J. Hwang. "A Tutoring Strategy Supporting System for Distance Learning on Computer Networks", IEEE Transactions on Education, Volume 41, Issue 4, November 1998.

[5] R. Anderson, R. Anderson, T. VanDeGrift, S. Wolfman and K. Yasuhara, "Promoting Interaction in Large Classes with Computer-Mediated Feedback", Computer Support for Collaborative Learning (CSCW), Bergen, Norway, June 2003.

[6] S. Brave, H. Ishii and A. Dahley, "Tangible Interfaces for Remote Collaboration and Communication", Computer Support for Collaborative Learning (CSCW), Seattle, Washington, pp. 169-178, November 1998.

[7] N. Faltin1, A. Böhne, J. Tuttas and B. Wagner, "Innovations in Virtual and Remote Laboratories", International Conference on Engineering Education, Manchester, United Kingdom, August 2002.

[8] J. Hua and A. Ganz, "A New Model for Remote Laboratory Education Based on Next Generation Interactive Technologies", ASEE New England Regional Conference, Orono, Maine, May 2003.

[9] Microsoft Windows XP Remote Assistance, http://www.microsoft.com/windowsxp/expertzone/focuson/remoteassist.asp, December 2003.

[10] Microsoft ConferenceXP Platform. http://www.conferencexp.net/community/, December 2003.

[11] A. Knight and K. Almeroth. "DeCAF: A Digital Classroom Application Framework", UCSB Technical Report, April 2004.

[12] K. Almeroth. "The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment", IEEE Network Special Issue on Multicasting, Volume10, Number 1, pp. 10-20, January/February 2000.