

An Online Tool for Capstone Design Course Communication

Loren Wyard-Scott
University of Alberta

The communication requirements associated with the instruction of a Capstone Design course can be extensive. This paper describes a utility used to simplify and improve communication between the instruction team members, and communication from the instruction team to the student design groups. This resource is part of a larger suite of tools that make use of a MySQL database to keep record of information. The communication tool is accessed through a web-browser connected to the database via a secure Apache web server and CGI scripts written using the Tool Command Language. The display of student identification photographs has allowed the instruction team to deliver personalized instruction even with large class sizes. This tool has been continually developed over the last five years using exclusively open-source or freely-available software and has greatly streamlined the communication required during instruction of Capstone Design courses.

Corresponding Author: Loren Wyard-Scott, wyard@ece.ualberta.ca

Introduction

Instruction of a large Capstone Design class is challenging in terms of communication requirements. To successfully deliver a course to a large number of students, effective communication between the instruction team members and the students is necessary. This paper presents select aspects of an online system developed to streamline administration and improve communication within the course.

At the time of project inception, existing class (learning) management and delivery mediums such as Moodle¹ and WebCT/Blackboard² were not consistent with several of the unique needs presented by instruction of a class in which group-oriented work of this nature was taking place. For instance, display of photographs of students while composing communication was not possible at that time. Additionally, the author took the position that learning to extend their functionality would be as daunting as creation of a new one.

Other tools, like the systems described above, offer desirable functionality. For instance, Lam and Maheshwari have created a Distributed Software Project Management Tool³ that runs on a proprietary platform and emphasizes a contained experience where no external email is required. Zhang et al have created a system⁴ that seems to emphasize the tracking of projects and also provides the students with a framework for communication. Another hierarchical web-based tool by Kanai⁵ is used to help faculty, students, and industrial clients keep in contact. This last system is developed on an open platform.

A common thread in all these systems is that they provide (and arguably emphasize) the role of students in their use. The author cannot deny merit to such an approach, and yet he also believes that an important part

of a student's learning experience lies in struggles such as selection of protocols and tools for things like group communication. Further, many students seem to prefer using communication methods to which they have grown accustomed.

The system described here is intended to provide a tool for the *administration* of Capstone Design courses, which includes communication between members of the instruction team, and communication from the instruction team to the students. Communication between the individuals in student groups is intentionally omitted.

The remainder of this paper is organized as follows. Context is first established in a section outlining the course format. Next, the system structure, of particular interest to those readers who may wish to implement a similar suite of tools, is presented. The main communication tool, termed the "Notes Server", is then described and is followed by a brief summary that includes subjective assessment of the described tool.

Course Format: Motivation

The framework described here has been developed to assist in the administration of a senior-year Electrical Engineering Capstone Design course but is appropriate for use whenever student teams are formed for the duration of a course. A typical run of the EE Capstone course involves the instruction of approximately 130 students who work in groups of four or five.

The instruction team consists of an academic instructor, a course technician, and four graduate teaching assistants (GTAs). The academic instructs the class in a lecture environment, is present much of the time in the laboratory, and keeps track of the progress of all groups. The course technician ensures the laboratory

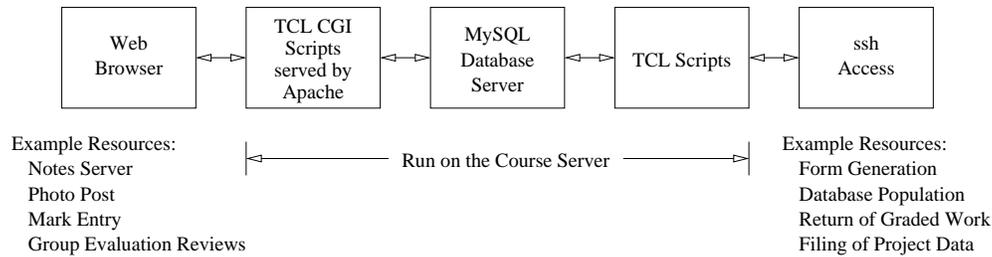


Figure 1 – A top-level view of the system structure. The contents of a database are accessed by scripts that format information for presentation to a web browser or a secure shell (ssh) connection.

environment is kept safe and operational, orders parts for the project prototypes, manufactures circuit boards and offers guidance to students. The GTAs are assigned a set of groups to pay particular attention to, but help all groups with design problems, troubleshooting and prototyping skills, etc. Depending on the nature of a project, groups are sometimes referred to other academics and technicians when unique domain-specific problems present themselves.

Effective communication between an instruction team’s members is important, particularly in a busy environment. For instance, if a GTA observes that a group is having difficulty with a topic, then the instruction team works more effectively if all members know about this: conversations can begin at appropriate levels, resources made available to the students, etc.

Finally, maintaining contact with a student group is very important in a design course. This can be time-consuming with traditional tools: the address of each group member needs to be located, entered into an email client, and the message composed.

The system described here helps reduce the amount of time communication and administration take. In particular, an online tool, available to all instruction staff via a web browser, helps meet communication needs. The next section describes the overall setup.

System Structure

As mentioned previously, the communication tool described in this paper is part of a larger set of communication and administration tools. This section describes the framework upon which the tools (also referred to as “resources”) depend.

The system is implemented on an Internet-connected PC running a distribution of Linux; the course server. The Internet connectivity allows all members of the instruction team to access the resources wherever network access is available. Many standard features of a Linux server are utilized to provide functionality; email transmission, for instance, utilizes the Simple Mail Transfer Protocol (SMTP). A top-level view of the system structure is shown in Fig. 1. This figure lists example resources that are made available through the

interfaces. This paper emphasizes the description of the “Notes Server”, listed on the left.

MySQL databases, structures of which are described below, contain information about the students, groups, the course, and the instruction team. This information is accessed via scripts written in the Tool Command Language (TCL), chosen due to the author’s experience. These scripts provide the bridge between either a web browser, via a secure connection to an Apache web server, or directly to a script run via a secure shell (ssh) session. In the former case, these are referred to as Common Gateway Interface (CGI) scripts. The resources accessed via an ssh connection are primarily those which only the course instructor requires access to, but will likely be adapted for web-based access in the future. Conversely, the web-based resources are intended for use by all instruction team members, and, for some resources such as electronic assignment submission, students.

Rather than having a single large application that implements all functionality, separate resources are implemented to maintain modularity. This separation additionally allows differing access privileges.

Database Structure

MySQL is used as the relational database server. Several databases are associated with the system, of which two are discussed.

The first database, *Access*, shown in Fig. 2, is used to provide instruction team members access to web-served resources. Each user belongs to a category that defines their role in the course: Administrator, or GTA. A TCL authentication script ensures that the user has permission to access a resource granted either by way of their category or explicitly via the *ResourceUser* table. Conversely, the script also ensures that a user is not explicitly denied access to a resource via the *ResourceUserDeny* table. This structure allows for very flexible specification of permissions. A marker GTA, for instance, may be allowed access to a resource used for mark entry, while lab GTAs are not. The *Resource* table provides the URL of the CGI script that delivers the resource. Options may be added to change the behavior of the resource. As an example, the

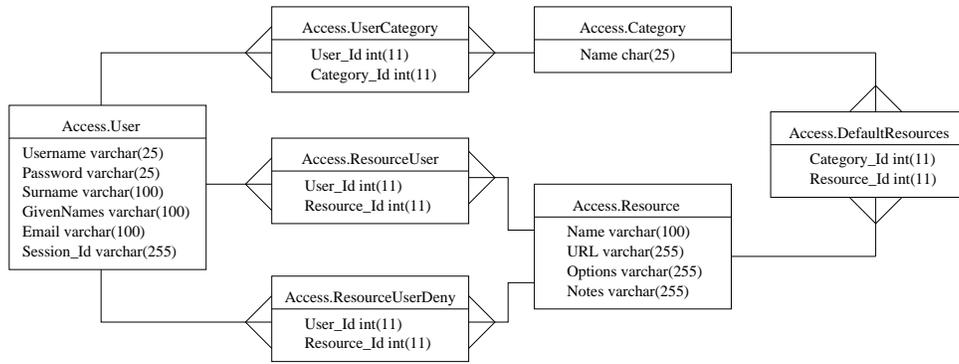


Figure 2 - The structure of the relational Access database governing instruction team access to the resources made available through the web interface.

“Notes Server” resource provides a slightly different interface for users in Administrator or GTA categories: administrators are permitted access to historical information by selecting a course term, whereas GTAs have access restricted to the current term. The same TCL script (and therefore the same base URL) is used to provide the interface for both categories of user, but the resources differ by one parameter specified in the Option field of the Resource table.

The second database is Student and contains information about the students and student groups, as shown in Fig. 3. The Student and StudentGroup tables are used by every resource. The Note table is used to keep record of observations made by the instruction team and emails sent to the project groups and individuals.

Other databases are used, but these are not pertinent to the description of the communication tool.

Communication Tool: The Notes Server

This section describes the features of the most refined of the web-based resources: the “Notes Server”. This resource offers the following capabilities to the instruction team:

- notes or observations may be made about

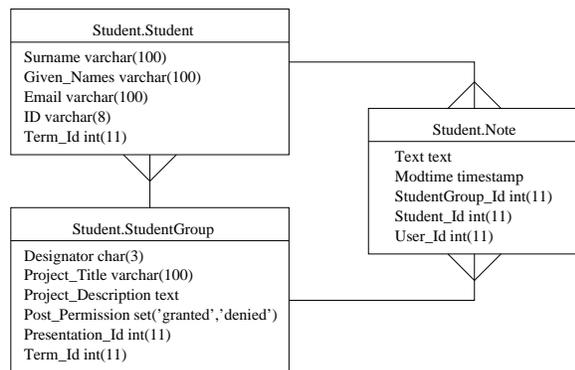


Figure 3 - The structure of the Student database.

individuals in a group or about the group as a whole;

- email messages can be sent to individuals or groups;
- administrative emails (messages pertaining to the group’s project but not for student consideration) can be sent;
- prior notes and email messages may be viewed;
- lists of students or groups can be accessed, with or without display of photos of students; and
- files associated with a group, including submitted and evaluated work, may be accessed.

Images of students are stored in the file space accessible by the web server and named according to the Student_Id field in the Student.Student table. Similarly, the files associated with a group are stored in a location consistent with the expectations of all resources that access these files.

Fig. 4 is a screen capture of an email being composed to a group (student photos and names are used with permission, but the content is fictitious and for demonstration purposes only). Once transmitted, the email is stored in the database and can be viewed by everyone in the instruction team. The links provided below the student photographs allow the system user to quickly jump to pages where notes can be viewed or made about the individuals, or email can be sent. Another link (not shown) allows for an email to be composed in an external email client. This provides a convenient way of sending files as part of an email, a feature currently not implemented in the Notes system.

Conclusion

The communication tool works very well. GTAs are asked to make daily entries about the groups they work with. The instructor makes extensive use of the system to send messages to students. The course technician

Select Term/Year
Logout

Project Group Notes: Create Email

View [Student List](#)

View [Group List](#)

Compose Email for Project Group

Designator: **15**
Project Title: **CNC Milling Machine Controller**
Number of group notes: [0 notes](#)
Group Members:



C., Jodi
[0 Notes](#)
invalid@ualberta.ca



L., Adam
[0 Notes](#)
invalid@ualberta.ca



R., Maysam
[0 Notes](#)
invalid@ualberta.ca



W., David
[0 Notes](#)
invalid@ualberta.ca



Y., Savio
[0 Notes](#)
invalid@ualberta.ca

Group Email Message

Subject:

CC:

(comma delimited)

Hi, Group 15:

The quadrature decoding firmware looks excellent.

Figure 4 – A partial capture of the web page showing the group email feature of the Notes Server resource.

utilizes the system to inform groups when parts arrive, special equipment is arranged for, etc.

In the current 2009-10 academic year, 1,222 notes have been made so far: 929 group notes, and 293 student notes.

Since the intention of each note is to keep all other instruction team members informed about student progress and difficulties, it can be equated to the transmission of an email message. Further, these notes can be considered to be sent to the instruction team. Assume that the overhead to compile an email message in a conventional client is 30 seconds (the time to enter the addresses of all recipients) and the time to file a message for future reference is 5 seconds. The transmission of 1,222 messages under these conditions would take an extra 18.7 hours of manpower to process, assuming an instruction team size of 6. Instead, this time can be used to provide face-to-face interaction with the students. Further time savings are realized with the use of further resources.

In addition to allowing the instruction team to quickly learn the names of students, the presence of photographs has virtually eliminated situations where emails have been sent to the wrong groups or students, or where the identity of students becomes confused.

Other resources built upon this framework add further functionality such as automatic form generation (via database queries and LaTeX-formatted input), acceptance of student peer evaluations as part of the autorating method, and acceptance and return of course work from students. Coupled with improved communication facilitated by the Notes Server, this suite of tools allows the instruction team to teach large Capstone Design class sizes in a time-effective manner.

Acknowledgements

The author would like to thank Jodi C., Adam L., Maysam R., David W., and Savio Y. for permission to use their photographs in this manuscript.

References

1. Moodle.org (n.d.). *Open-source community-based tools for learning*. [Online]. Available: <http://moodle.org>
2. Blackboard Inc (2010). *Blackboard Home*. [Online]. Available: <http://www.blackboard.com>
3. H.E. Lam and P. Maheshwari (2001). "Task and Team Management in The Distributed Software Project Management Tool". In *Proceedings of the 25th Annual International Computer Software and Applications Conference*, pp 401-408.
4. J. Zhang, D. Zage, and W. Zage (2003). "Improving Project Planning/Tracking for Student Software Engineering Projects through SOPPTS". In *Proceedings of the 16th Conference on Software Engineering Education and Training*, pp. 185-192.
5. J. Kanai (2008). "Web-Based Collaboration Tool in Engineering Design Courses". [Online]. ASEE Zone 1 2008 Professional Papers Proceedings Archive. Available: <http://www.asee.org/activities/organizations/zones/proceedings/zone1/2008Professional.cfm>