

Beyond Clickers: Using ClassQue for Multidimensional Electronic Classroom Interaction

Steven Robbins
Department of Computer Science
University of Texas at San Antonio
srobbins@cs.utsa.edu

ABSTRACT

ClassQue is a classroom response system that goes beyond clickers to allow a wide selection of classroom interactions: teacher to individual student, teacher to all students, student to teacher and student to student. Questions are not restricted to multiple choice, and multiple questions can be pending at one time. One student can anonymously comment on another student's answer. After the class, students and teachers can receive reports of the classroom interactions. The current version of ClassQue is available for use in an environment in which each student is seated at a computer.

Categories and Subject Descriptors

K.3.2 [Computers & Education]: Computer & Information Science Education—*Computer Science Education*

General Terms

Management

Keywords

Classroom Response System, Clickers, Curriculum

1. INTRODUCTION

Much has been written about how the traditional teacher-student interaction through lectures has failed to keep student's attention [4]. Producing an environment in which students are willing to interact with the teacher and classmates can be difficult, especially in a large class or in a class in which the students feel unsure of their abilities. Clickers [1, 2, 3] have been used for several years to allow teachers to pose questions to students and get real-time response information. Responses can drive or motivate classroom discussion and pique student interest. Others have used graphing calculators [9] or tablet PCs [7].

In a typical clicker application, a teacher asks a multiple choice question, usually by posting it using an overhead projector. The students then have a fixed amount of time to answer. When the time expires, the teacher displays and discusses the question. The

teacher must allow enough time for the slowest students in the class to answer the question. Meanwhile, there is dead time in which students who have answered the question must wait quietly for the full time to expire.

The dead time problem is addressed by Informa [5, 6], an extensible system written in Java that has students use standard computers in place of clickers. An Informa plugin includes both a question type and an aggregate visualization for displaying solutions. Informa has been used to teach introductory Java programming, with an enhanced set of question types. Informa addresses the issue of dead time by allowing students to evaluate other student's answers. In the Informa terminology, each question has three phases: solve, evaluate, reveal. In the solve phase, a question is posed and students work on the solution. During the evaluate phase, students who have solved the problem evaluate the solutions of other students. In the reveal phase the teacher displays solutions and discusses the results.

ClassQue, which also uses computers in place of standard clickers, provides a wide selection of question types that are applicable to many teaching situations. ClassQue fills the dead time with student reviews, allowing multiple questions to be pending, and allowing questions with open-ended answers. After answering a question, a student can look at and comment on other students answers to that question, or look at reviews to his/her answers. Teachers can also review or grade student answers.

An important aspect of clicker responses is the impression of anonymity they give. This allows students to answer freely without fear of scorn or intimidation from classmates. Having responses mandatory (in the sense that a lack of response can have a negative impact on the student's grade) is also useful in encouraging students to respond. In ClassQue, students do not know which students they are reviewing or which students have reviewed their answers.

During the discussion phase, teachers can display all of the answers given by the students in several formats including anonymous summaries and can also show reviews given to the answers.

ClassQue has support for students requesting help from the teacher or a remote teaching assistant without disturbing the rest of the class. It also supports group communication among students.

ClassQue requires the teachers and students to have a network-connected computer capable of running a recent version of Java. It is best used in an environment in which the teacher can display a computer screen using an overhead projector. The ClassQue system is available for download [8].

The rest of the paper is organized as follows. Section 2 discusses the basic ClassQue interface from the student and teacher viewpoint. Section 3 describes additional ClassQue features. Section 4 discusses a utility for creating questions for use by ClassQue. Section 5 goes into the details of the operation of ClassQue and Section

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

6 presents some conclusions.

2. CLASSQUE BASICS

When students enter the classroom they sign in to ClassQue which then displays an initial window. Figure 1 shows a student window after a simple query has been issued.

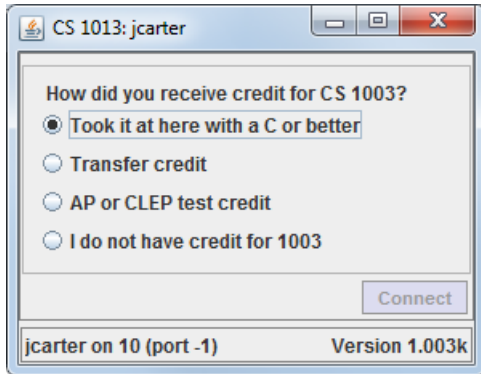


Figure 1: A student query window.

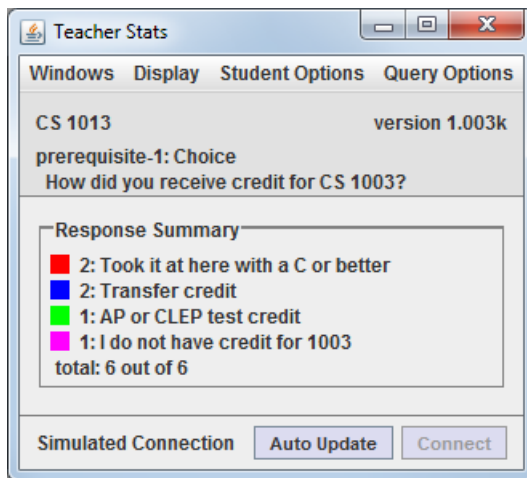


Figure 2: A teacher summary window.

ClassQue supports five basic types of queries. **Choice** queries allow students to choose one of a predetermined selection of answers. **Checkbox** queries are similar, but students can choose more than one answer. **Integer** queries allow students to enter a single integer as a response. **Freeform** queries allow students to make arbitrary text responses ranging in size from a few words to many paragraphs and possibly pasted from another document. **Code** queries allow student to type in program code as a response, displayed with a fixed width font and preserving indentation and spacing.

Figure 2 shows an example of a response summary for the multiple choice query of Figure 1. This display, which is similar to those of clicker systems, is one of the many possible teacher display formats. The **Responses by Student** view of Figure 3. displays a list of the students who have answered the query along with the answer given. The **Responses by Answer** view gives a list by answer, with the count of the number of students giving that answer. The underlined values in the responses shown in Figure 3 are links. Clicking on a **Review** link brings up a review window (described in Sec-

tion 3.2). Clicking on a name or numbered answer link shows or hides the reviews for that answer.

Figure 4 shows a detailed view of the answers to a freeform query. In this figure the answer for each student is shown in the order that the answers were given, and the teacher has chosen to hide names of the students, with the name replaced by a number in square brackets.

Figure 5 shows two views of the answers to a query with an integer answer. The **Responses by Student** view shows the response of each student. The teacher has the option of hiding the name of the student giving the answer, so that this window can be shown to the class for discussion. The **Responses by Answer** view is organized by answer. The value after the colon is the number of students giving the answer that appears below.

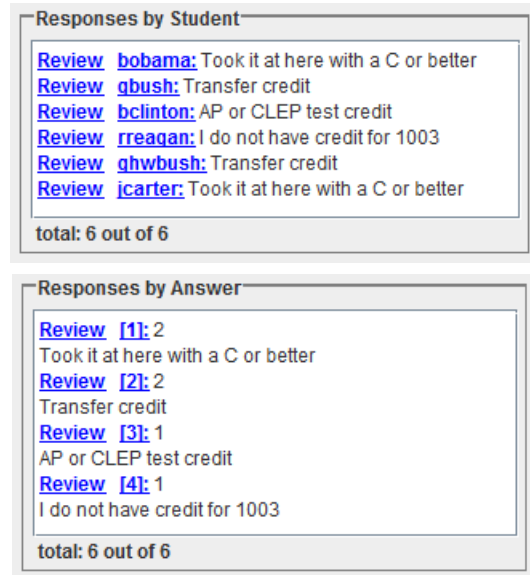


Figure 3: Two additional teacher views for a query with a choice answer.

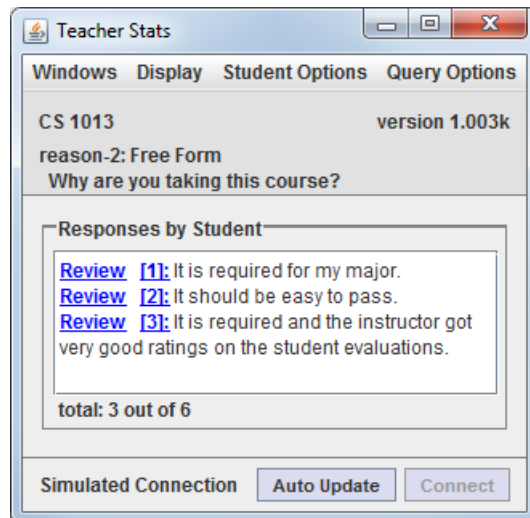


Figure 4: A teacher window for a query with a freeform answer.

Clicker technology requires students to work in lock-step with the slowest members of the class. Students who answer quickly are

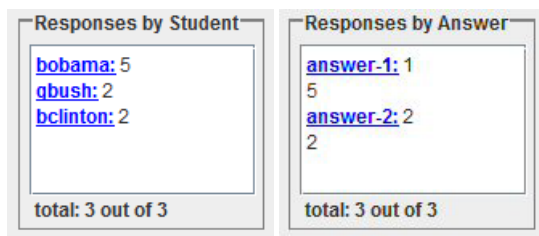


Figure 5: Two teacher views for a query with an integer answer.

left with nothing to do until the answer time has completed.

ClassQue handles this problem in two ways. First, ClassQue supports the use of surveys. A survey is a collection of queries that students can answer at their own pace. The queries do not have to be answered in order. Some of the queries can be optional, or have open-ended answers, allowing the faster students to answer more queries or give more in-depth answers.

ClassQue also gives students something to do after they have answered all of the pending queries. If the teacher allows it, students can look at individual (anonymous) answers that other students have given and comment on those answers. These comments (referred to as student reviews) are also sent to the student that gave that answer. Students can see these comments, again anonymously, and have the option of adjusting their own answers. These revised answers can also be reviewed.

While waiting for responses, the teacher can also review student answers. Teacher reviews are marked, so students can see that these reviews are authoritative. Teaching assistants can also review answers. Teaching assistants do not need to be physically in the same classroom, but can log into a ClassQue assistant console from any machine on the network.

The teacher can also use the wait time to decide which of the student responses to use as a basis for classroom discussion. During the discussion, the teacher can display a window similar to Figure 6 and scroll through all of the answers, asking for comments or quickly describing what is right or wrong with each answer. The teacher can hide the names of the students to avoid embarrassing students who have given incorrect answers. This is important for encouraging student to submit answers.

3. CLASSQUE FEATURES

The ClassQue system consists of a suite of Java programs. During class, students and teachers communicate with a server. Typically, the server program starts automatically before each class (e.g., as a cron job on a Linux system) and the server terminates after the class is over. While the server is up, students, teachers, and assistants can sign in. The server keeps track of all queries, answers, and reviews, logging them to a file on the server host. After the class is over, reporting programs automatically process and email reports to the teacher and the students. Teachers can also receive attendance summaries.

3.1 Seating Chart

Figure 7 shows two teacher's seating chart views. ClassQue supports seating diagrams consisting of one or two rectangular arrays of seats. This seating chart view is useful for quickly picking out which students have answered the current query. The left diagram in Figure 7 shows answers for a **Choice** query. The color of the box indicates which answer was given. Color codes correspond to the colors in Figure 2. The right diagram shows answers for a **Checkbox** query. If more than one answer is selected, the box is

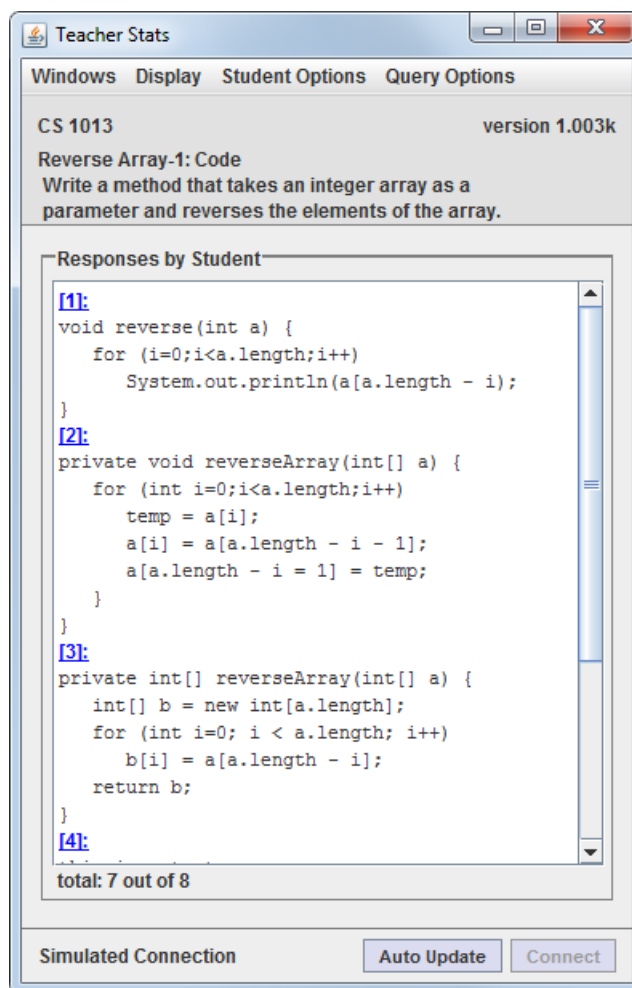


Figure 6: A teacher window showing many solutions to the same problem.

divided into regions. For example, in the right diagram of Figure 7, the student in seat 1 selected two answers (red and blue) and the student in seat 26 selected three answers. In all cases, a light gray box indicates that the student is signed in but has not yet answered the query.

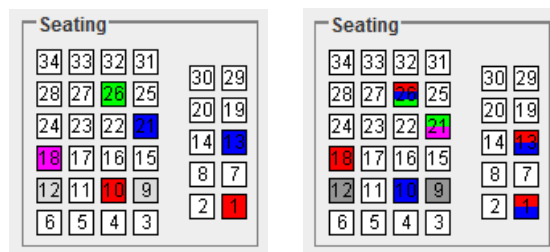


Figure 7: Seating charts for choice and checkbox queries.

3.2 Reviews

ClassQue allows students or teachers to comment on (review) the answers given by students. A teacher can review an answer by clicking on the **Review** link optionally displayed in the Responses

window shown in Figure 4. This brings up a review window containing the query and the student answer along with an area for commenting on the answer given.

Teachers can also allow students to review the answers of other students on questions they have already answered. When a student requests an answer to review, ClassQue anonymously chooses a response using a method that attempts to maximize the number of answers that are reviewed by students. A student can see the reviews of his/her answers. Teacher reviews are marked, and student reviews are anonymous. After seeing a review, students can revise answers and resubmit.

If a query is of type choice, checkbox, or integer, it is likely that several students will give the same answer. In this case a review is associated with the answer, so that once an answer is reviewed, all students who gave that answer can see the review.

Teachers can also display **Correct** and **Incorrect** links in the **Response** area to quickly mark answers as right or wrong.

Figure 8 shows a portion of the teacher's response window in which some reviews are shown. Student reviews are anonymous, but reviews by the teacher or assistant are marked as such so that students can pay careful attention to these.

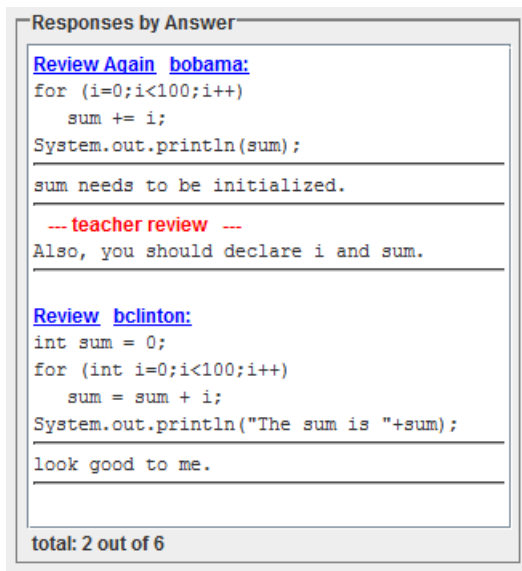


Figure 8: A part of a teacher window showing reviews.

3.3 Shares

Shares allow groups of students to share information and communicate. Suppose the teacher asks the students to collaborate on developing a method with a given specification. Each student in the group will have a *common window* for typing in. All students belonging to the share see the same contents in the common window and can contribute.

The share also has a chat area that allows students to post comments that other members can see. While the contents of the common window are unattributed, the chat area shows a history of the attributed comments made by each member of the share. When the students agree that the solution is complete, a member of the group can submit the contents of the common window. This saves a permanent copy of the contents of the window, which the teacher can later grade. Students can move the contents of the shared window into another program using cut and paste.

Figure 9 shows the student view of a share with the common win-

dow near the top and the chat session below for writing a method to reverse the elements of the array. The figure shows the share for **bcClinton**. Other members of the share are **gbush** and **bobama**. Most of the writing was done by **bcClinton** but the other two members made contributions as shown in the chat area.

A teacher can join any share. As with any other member of the share, the teacher can add comments to the chat area or modify the common window. The teacher can also add or remove members of the share or submit all shares.

ClassQue allows students to create their own shares and invite students to join that share. ClassQue allows teachers to easily divide the students into groups, either by explicit choice or by random selection and to create a share for each group. Each share has a leader who has priority in writing to the shared window.

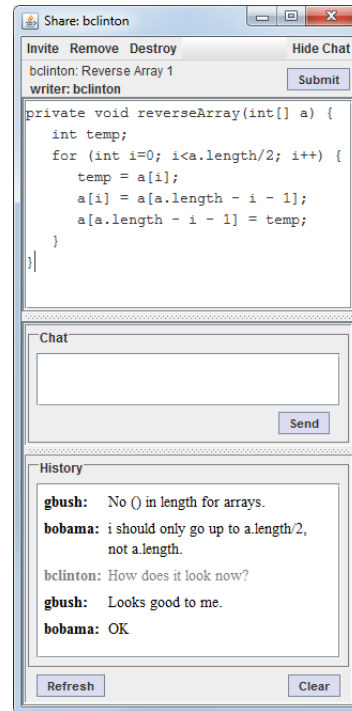


Figure 9: A student share window (reduced in size).

3.4 Messages

A teacher can send a message to an individual student or to a group of students. Figure 10 shows the window that a teacher can use to send a message to students. ClassQue allows two types of messages, those that require a response and those that do not. The figure shows a message sent to three students, two of whom have replied. The teacher can display the reply by hovering the mouse pointer over a student name.

Messages immediately appear in the student console window if the student is signed in. Otherwise they are queued until the student signs in, allowing the teacher to send before the class starts. I have used these messages to congratulate students who have done particularly well on an assignment or have put in an extra effort, and these have evoked positive responses from the students. Messages are also useful to obtain clarifications on assignment solutions that students have turned in.

3.5 Passwords

ClassQue can be set up to require a password from the teacher.

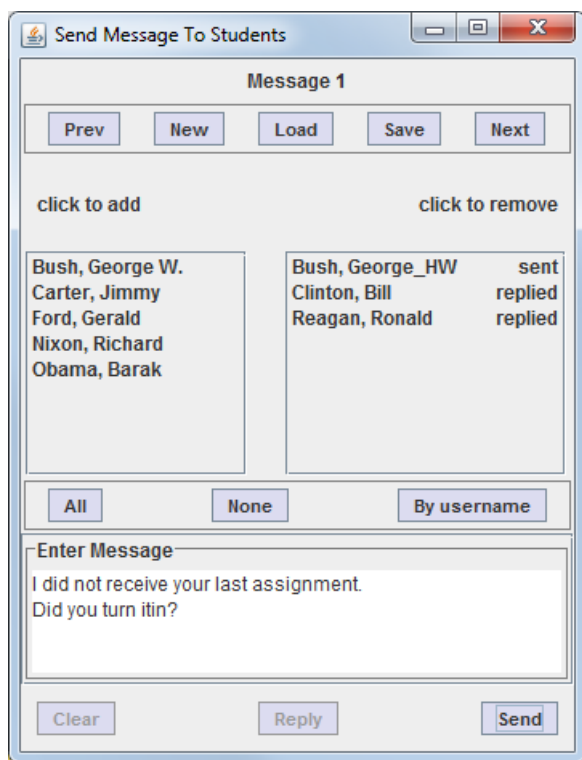


Figure 10: A window for creating messages.

Without passwords, anyone with a copy of the teacher program can run as a teacher and disrupt the ClassQue system by injecting extraneous queries or sending messages to students. We ran the ClassQue system for a year without passwords and never had a problem with this. However, the public release of ClassQue will make it easier for hackers to disrupt a class if passwords are not used.

Students who sign in using the ClassQue student application do not have to enter a password, but they are identified by their host machine. ClassQue can be set up to allow only students from a specified collection of IP addresses to discourage hackers from disrupting the system. Students who log in using the browser interface are required to enter a username and password.

3.6 Other Features

Student List: The teacher can display a list showing where students are or should be sitting and whether they have signed into ClassQue. A teacher can also use this feature to sign students out of ClassQue, for example, disconnecting students who post inappropriate answers.

Pictures: If student pictures are available, ClassQue can display pictures of the students in a chart corresponding to where the students are sitting. This feature is useful for learning the names of the students in the class.

Help: Teachers can activate a student **Request Help** button. Students use this button to notify the teacher or an assistant that they need help. This feature is useful when the class is working on a classroom exercise. The teacher or an assistant can view a list of students who have requested help and provide assistance. The student does not have to wait for teacher attention with his/her hand raised.

The Assistant: ClassQue can be set up with two types of passwords for the teacher. The *teacher password* gives the user full teacher

privileges. The *assistant password* marks the user as an assistant with restricted privileges. An assistant can sign in from one of the computers in the classroom and be available to offer physical assistance to students who request help or be signed in remotely and offer assistance using ClassQue messages (Section 3.4). In either case, assistants can review student answers. Reviews given by the assistant are marked as such, but are still distinguished from student reviews.

Annotations: A teacher can add annotations to the queries. An annotation can be attached to the query (the prompt) or to an individual answer of a choice or checkbox query. Annotations are not seen by the student when the queries are asked, but can be seen by the teacher. Annotations can be used to indicate the correct answer to a query, or to provide additional feedback to students in the reports they receive.

Start Discussion: When the teacher pushes the **Start Discussion** button, the time is logged. This marker indicates that the teacher has started discussion of the current query, so that answers given after this time can be ignored by the reporting system or treated as additional notes.

Greeting: The teacher can set up a file with a message, allowing instructors to insert instructions at the beginning of the student reports. Instructions might include the assignment for the following class or a reminder of due dates.

Simulator: The simulator is a standalone program that allows instructors to run the full system on a single machine. The simulator, which includes a server, teacher windows, and student windows, can be used for training of instructors, testing questions and verifying the seating chart. Teachers can use the simulator to read in log files from a previous class to recreate the state of the ClassQue system from any point in that class. This capability allows instructors to perform additional operations such as reviews after the class has ended or to send a student an individual message that will appear in the student's report.

4. THE QUERY CREATOR

Queries can be generated on the fly during class, but the usual way is to have a prepared set of queries in a file. Normally, the teacher loads this file when he/she signs into ClassQue at the start of class. In order to allow students to start answering queries before the class starts and before the teacher arrives, ClassQue provides a mechanism for starting a teacher client on a remote machine in such a way that it will wait until the server starts up and then send the queries.

The file format used for ClassQue queries is a user-readable ASCII text query file, but the structure is fairly complex, and it would be difficult to create one of these files manually. ClassQue comes with a **Query Creator** program for creating and editing these files. The Query Creator is menu driven, allowing the user to create a query by choosing the query type, entering the prompt, and if appropriate, the choices for the answers. The Query Creator also allows for entering annotations.

ClassQue supports queries formatted in HTML to allow for colors, font changes, and even tables. The ClassQue Query Creator provides an interactive environment for creating, editing, combining and previewing query files.

5. SETTING UP CLASSQUE

Download ClassQue [8] and unzip the file. Among the files included will be jar files for the server, teacher, student, query creator and report generators as well as a user's guide.

Each course that uses ClassQue has two port numbers dedicated

to it. The ClassQue server is a Java program that can run on any system that supports Java 6. The server hardware and operating system needs to be able to handle multiple connections in a timely manner. A three-year-old mid-level Linux desktop can easily handle a couple of classes of 40 or more students simultaneously, without noticeable delay.

A copy of the ClassQue server is started before each class, and it uses one port to communicate with the teachers and one to communicate with the students. The server terminates after running for a specified length of time. For example, if a class meets MWF from 9:00 am to 9:50 am, the server might be started at 8:50 am every MWF by a cron job on a Linux machine. A configuration file provides the port numbers for this copy of the server as well as specifying how long the server runs before termination. For the above class, the duration might be set to 70 minutes, which will allow it to run until 10:00. All communication among teachers, assistants, and students flows through the server.

In addition to handling communication, the server logs sufficient information to recreate the ClassQue state at any time. One of the log files generated is an XML file that logs all teacher queries and student answers, as well as reviews and submitted shares. After the server terminates, another cron job can start a report generator that analyzes the XML log file and sends reports to the teacher and the students.

The teacher and student programs are clients of the server. They can be run on any system that supports Java 5 or later. Alternatively, the student console can be run from a browser. If run as an application, the student console attempts to determine the username and seat number from the host machine. If running from a browser, the student logs in, giving his/her username and seat number. In either case, a given student may run only one copy of the program at a time.

The teacher may have any number of sessions running. Multiple sessions on the same machine allow for several views of the student information. Additional teacher client programs can be run on other hosts. A special type of teacher, called an assistant, behaves like a teacher, but with restricted privileges. This can be used by a teaching assistant to interact with the students. The assistant does not need to be in the classroom.

6. CONCLUSIONS

ClassQue has been used for three semesters by 6 different instructors. It has been used in courses in MATLAB programming, Java programming, computer organization and operating systems.

While no comparison was made between courses using and not using ClassQue, ClassQue was well-received by the students and instructors. Students found it particularly useful in the operating systems class in which a survey was given at the beginning of class to evaluate the comprehension of the previous class. These results were mainly used as formative feedback, highlighting parts of the course material that needed further explanation. The instructor posted the queries (with answers) on the web, and students found this helpful in reviewing for exams. In some cases the use of ClassQue seems to have had a positive effect on the student evaluations of the course.

We teach our beginning programming courses using a development system such as Eclipse. Some faculty members believe that it is better for students to develop programs on paper without the assistance of a system that automatically finds and corrects syntax errors. Programming on paper forces students to think carefully about the programs they write, rather than to program using trial and error. We have used ClassQue shares (Section 3.3) to force students to first write their programs in a non-development environment and

then paste them into Eclipse for debugging.

We have also used ClassQue in modified pair programming exercises, which I call barrier pair programming. Using traditional pair programming in a classroom situation suffers from a number of technical difficulties: including dividing the class into pairs, moving students around, making sure that a stronger student does not dominate a weaker one. For short problems, the overhead of setting up this environment makes it impractical to use in a 50-minute class period.

Barrier pair programming handles the logistics as follows. We design a problem with fixed milestones and use ClassQue to randomly divide the class into groups of size 2. Each student in the class works on the problem independently. When a student reaches a milestone, that student may not proceed (the barrier) until the paired student has also reached that milestone. The student who gets there first must help his/her partner (using the ClassQue share). A ClassQue checkbox question with the milestones as choices can help the teacher monitor the progress of the class.

ClassQue is available [8] for download. The single zip file contains jar files for the server, student, teacher, query creator, and report generators. It also includes a user's guide.

7. ACKNOWLEDGMENTS

Kay Robbins has written the report feature of ClassQue and has provided numerous useful suggestions.

8. REFERENCES

- [1] A. L. Abrahamson, "An Overview of Teaching and Learning Research with Classroom Communication Systems," *International Conference on Teaching of Mathematics*, (1998), Village of Pythagorion, Samos, Greece.
- [2] S. Draper, J. Cargill, and Q. Cutts, "Electronically Enhanced Classroom Interaction," *Australian Journal of Educational Technology*, (2002), pp. 13–23.
- [3] C. Fies and M. Jill, "Classroom Responses Systems: A Review of the Literature," *Journal of Science Education and Technology*, (2006), 101–09.
- [4] R. R. Hake, "Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *American Journal of Physics*, vol. 66, (1998) pp. 64–74.
- [5] M. Hauswirth and A. Adamoli, "Informa: An extensible framework for group response systems," *Proceedings of the 4th International Conference on Collaborative Computing*, (2008).
- [6] M. Hauswirth and A. Adamoli, "Solve & Evaluate with Informa: A Java-based Classroom Response System for Teaching Java," *Principles and Practice of Programming in Java*, (2009).
- [7] K. Koile and D. Singer, "Development of a Tablet-PC-based System to Increase Instructor-Student Interactions and Student Learning," *Workshop on the Impact of Pen-based Technology on Education*, April 2006.
- [8] S. Robbins, ClassQue, 2010, Online. Internet. Available WWW: <http://vip.cs.utsa.edu/classque>.
- [9] J. Roschelle, P. Vahey, D. Tatar, J. Kaput, and S. Hegedus, "Five key considerations for networking in a hand-held-based mathematics classroom," *Proceedings of the 27th Conference of the International Group for Psychology of Mathematics Education*, (2003).