

The Design, Implementation, and Application of the GrewpEdit Tool

Kenroy Granville, Timothy J. Hickey

Abstract

This paper examines the GrewpEdit tool which was developed as part of the Groupware Research in Education and the Workforce Project. GrewpEdit was designed to support close collaboration in programming classes. We describe the design and use of GrewpEdit and discuss our current and planned uses of this tool.

1 Introduction

Over the past five years we have been involved in building and analyzing same time, different place software to support collaborative programming in undergraduate Computer Science courses. This work was partially funded by the NSF and has resulted in several systems. The TATool [4] was our first tool; it allowed a student and TA to see each others' text, but each could only modify their own code. The next two versions were the GroupHomeworkTool (GHT) [5] and the GREWPTool [2, 3]. These provided a more sophisticated interface which allowed many students to simultaneously edit a single document as well as chat, co-browse the web, share a whiteboard, etc. The GREWPTool also had a logging/playback feature in which a session could be logged to a file and then played back and analyzed using a VCR-like interface [1]. These tools were research platforms and were not designed to be widely disseminated and used in classrooms. They were built to allow researchers to study collaboration.

This year, we have focused on building a collaborative editing and programming tool called GrewpEdit that can scale up to support hundreds of users working on hundreds of documents, and that is easy for teachers, teaching assistants, and students to use. This paper will focus on our experience with and plans for the GrewpEdit tool.

2 GrewpEdit Features

The features of GrewpEdit have expanded substantially over the four generations of groupware tools we have produced for this project. The most significant additions built into GrewpEdit are an improved rendezvous method (for arbitrary group sizes independent of location) and a plugin-like architecture for distributing a diverse set collaboration tools using a common rendezvous/awareness interface.

2.1 Rendezvous

When a user launches the GrewpEdit application they are automatically connected to our default RendezvousServer and greeted with the Login Window (see Figure 1). Because the default login window simply contains fields for a username and groupname, a list of existing groups, an area to report information about the selected existing group, and a "GO" button, joining or starting

a group takes minimal understanding. The only overt action to the user is the "GO" button and it either starts or joins a GrewpSession with the username and groupname in the fields. To determine whether to start or join a GrewpSession, the RendezvousServer checks the uniqueness of the groupname against Existing Groups, thereby starting a new GrewpSession if verified as unique and joining the corresponding group otherwise. The username is also subjected to verification and alteration is requested in failure.

2.2 Grewplets

The Grewplet idea introduced in GrewpEdit provides a mechanism for easily creating new Groupware applications that can automatically make use of a shared Groupware platform, including rendezvous, user awareness, and application launching interfaces. Our solution is a single system that provides a central base, yet is capable of supporting multiple forms of collaboration.

2.3 Deployment

GrewpEdit was developed using JScheme, an implementation of Scheme in Java (<http://jscheme.sf.net>). GrewpEdit is a cross-platform application that has been tested on several different versions of Windows, Mac OS X, and Linux platforms. It is a part of the open source project groupscheme hosted at <http://groupscheme.sf.net> with a free open source license. Those interested can launch it as a Java Web Start application from the main website (<http://groupscheme.sf.net/grewpedit>) or download it as a jar file to be executed without Java Web Start. To promote further research and community development we make the source available for download from the host site with instructions on how to build the application.

Lets now walk through an example of a typical online class session.

It would start with the instructor launching GrewpEdit before class begins and entering "Instructor" as the username, "CLASSNAME" as the groupname, and clicking "GO". At this point the instructor can decide to make it a private group with a password or simply leave it public.

As students launch GrewpEdit, "CLASSNAME" will be in the list of Existing Groups which appears in the login window. When a student selects "CLASSNAME" in the list, it will be filled into the groupname field and the number of people already in the virtual classroom will be displayed in the area at the bottom, along with other information. After entering their name as username, and clicking "GO" to join the class GrewpSession, the student would be prompted for a password before being allowed to enter, if the GrewpSession was password protected, if not, he/she would be brought directly into the session.

For more knowledgeable users, we provide Rendezvous and Advanced Login options in the menu where a user could host their own or connect to a specific RendezvousServer and host or connect directly to a GrewpSession.

After starting and/or joining a GrewpSession, a user is presented with a GrewpSession Window that initially only contains the GROUPNAME tab (see Figure 2). This tab is the central collaboration point where students and instructor can communicate using the IM-like chat at the bottom. They can send chats to the whole group or to a specific user, which could be useful when a teacher is helping a struggling student privately.

As with the Login Window, we tried to minimize the number of GUI afforded actions by only placing three buttons on the tab, Join, New, and Change Color. New is directly connected to the Grewp Types list (a list of loaded Grewplets located at the top, right area of Figure 2) as it will start

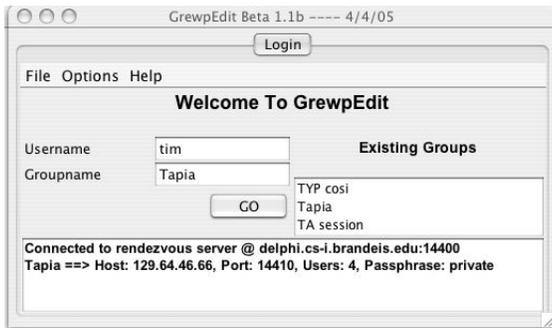


Figure 1: Rendezvous windows



Figure 2: GrewpSession windows

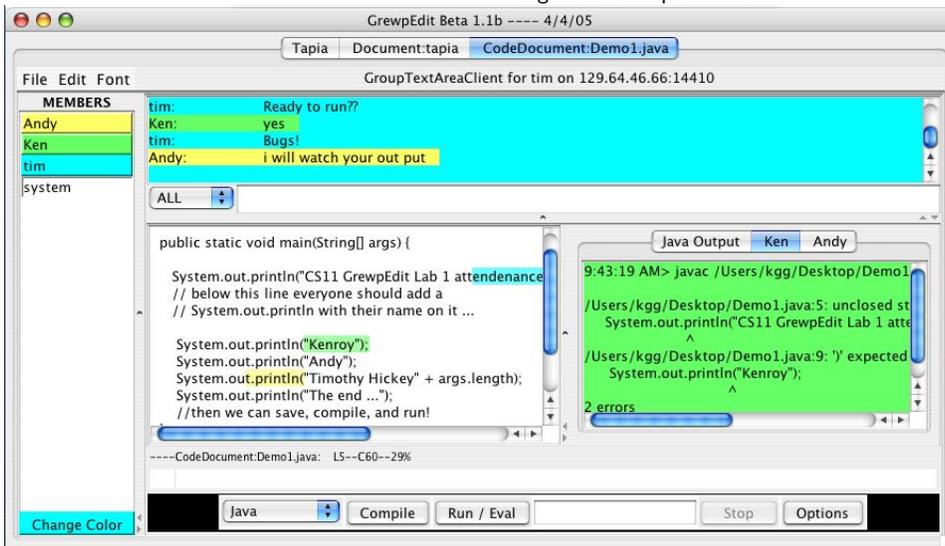


Figure 3: GrewpDocument window

a new Grewplet of the selected Grewp Type. It then appears as another tab in the GrewpSession Window (see Figure 3) and appended to the Grewp Places list (middle area of Figure 2) for other members to join.

When all the students have joined the GrewpSession the instructor starts a new CodeDocument and names it Demo1.java (see Figure 3). While this only adds a new tab, containing the Grewp Place, on the Instructor's GrewpSession window, all students will now see the corresponding name in the Grewp Places list. Following the instructor into the new Grewp Place, the class writes a simple Java program with aid supplied by the instructor. When the students feel they have completed the program, they press the Compile button only to receive an error message in the output area (middle, right area of Figure 3). Each person's output appears in a separate tab which everyone can see. This will allow instructors to help students debug programs using the students compiler/runtime system.

3 Future Work

The development direction of GrewpEdit focuses on refining the existing application to become more robust, expanding the architecture to include Grewplets written completely in Java (as it currently only supports JScheme), designing more Grewplets, and adding useful features, within limits.

A must add feature is logging/playback enabled Grewplets that log a transcript of all actions for VCR-like playback at a later time. In hopes to better support slower connections (dial up) and extremely long-distance coediting in our shared- editor Grewplet, we also plan on implementing a speculative concurrency control algorithm that will hopefully alleviate some of the lag time that appears in groups that are several thousand miles apart.

We feel that the GrewpEdit approach could be used to deliver educational resources (e.g. teaching assistants and instructors) more effectively than is now possible. Currently, students who want to get help with writing, running, and debugging programs must physically come to a teaching assistant or instructor. With GrewpEdit, they can get access to an instructor directly from their home or school computer. This could be used to establish college/high school collaborations where college students would act as online teaching assistants in selected high schools. The students on both sides would gain valuable experience. We are currently using this approach to provide online Teaching Assistant services to undergraduates at Brandeis and are planning on reaching out to selected high schools next year.

References

- [1] Svetlana Taneva, Tim Hickey, Richard Alterman, Collaborative Learning: Collaborative Depth, Brandeis Computer Science Tech. Report, CS-05-256, 2005
- [2] Timothy J. Hickey, J. T. Langton, Kenroy Granville, Richard Alterman, Enhancing CS Programming Lab Courses Using Collaborative Editors, Journal of Computing Sciences in Colleges, volume 20, number 3, Feb 2005.
- [3] Svetlana Taneva and Richard Alterman and Kenroy Granville and Michael Head and Timothy J. Hickey, GREWPTool: A System for Studying Online Collaborative Learning, Brandeis Computer Science Tech Report, CS-03-239, 2004.
- [4] Timothy J. Hickey, J. T. Langton, Kenroy Granville, Richard Alterman, TA Groupware, Tech. Rep. CS-02-222, CS Dept., Brandeis University, 2002.
- [5] John Langton, Richard Alterman, Timothy J. Hickey., Integrating Tools and Resources: a case study in building educational groupware for collaborative programming The Journal of Computing Sciences in Colleges, 19(5), May 2004.