Educational Fusion

Tools for Teachers in the eduFuse System

6.199 - Advanced Undergraduate Project
Submitted to Prof. Seth Teller
By Bhuvana Kulkarni

May 15, 2000

Abstract

Educational Fusion, or eduFuse, is a Web-based system written in Java that provides a visually oriented, collaborative learning environment. This system has a two-fold goal of making the pedagogical process more efficient for both students and instructors. First, eduFuse aims to make students' time more efficient by abstracting the details of the programming environment, which enables students to focus on the algorithmic concepts being taught. Second, the system hopes to reallocate teachers' time to include fewer administrative tasks, which creates more time for interaction & collaboration with students. The eduFuse system has been in existence since 1996; in this time numerous features have been designed and implemented that achieve the goal of making the process more efficient for students. However, there are only a few features in the eduFuse system that have been designed to fulfill the goal of making the process more efficient for instructors. Therefore, to extend the features for teachers within eduFuse, I have designed and implemented an online grading form utility.

1. Overview of eduFuse

1.1 Motivation for eduFuse

In recent years, the computing industry has experienced an explosion of activity on the Web as people all over the world are increasingly interconnected through millions of networked computers. The eduFuse system aims to take advantage of this worldwide network by providing a learning environment in which users all over the world can learn from and collaborate with each other without being physically close to one another [Boy99]. These ideas, termed distance learning and online collaboration are gaining more momentum, as many people are eager to learn from renowned teachers in various fields of study who may be located all over the world. Distance learning and online collaboration are especially useful for those who are studying or teaching at a university level, mostly because numerous users in university communities often have high speed, high bandwidth network connections.

Therefore, most of the users of eduFuse are university-level students and instructors. The eduFuse system is encapsulated in a Java applet; this makes it possible for users to access their work within eduFuse from any computer with a Java enabled web browser [Boy99]. In addition to the virtually unlimited access offered by eduFuse, the system also contains a robust set of communication tools that provide seamless collaboration functionalities to all users [Ken99]. The eduFuse system further makes use of the vast network of computers by providing server side compilation. With this feature, a user who is accessing the eduFuse system from a client browser does not need a compiler installed for a development environment; the only necessary software is the Java Virtual Machine (JVM) that is provided in most Java-enabled web browsers [Boy97].

1.2 Goals of eduFuse

As discussed above, the goals of eduFuse are dual; the system hopes to make the pedagogical process more efficient for both students and teachers [Tel94]. Traditionally, in university level classes with a programming component, students who are starting the first programming assignment must spend some time learning how to use the development environment [Por98]. This environment can include a compiler, interpreter, debugger and other tools. Many instructors would prefer that students could avoid this overhead time spent learning how to use the development environment, and instead could immediately start learning key algorithmic concepts. Many of the features of the eduFuse development environment, including the abstraction of details of tools in the development environment and the visual representations of algorithmic concepts, have been designed and implemented with this goal in mind.

The second goal of the eduFuse system is to make teachers' time more efficient. When instructors are using traditional teaching methods, a significant portion of their time is used for administrative tasks. These tasks include designing a curriculum, creating new assignments, and collecting & grading students' completed assignments. Many university level instructors have assistance with these tasks from teaching assistants; however, online tools within the eduFuse system that allow instructors to spend less time on administrative tasks would be very useful. One such tool in the eduFuse system that can help instructors is the ease of module creation [Por98]. Another tool that has been recently added to the system is the online grading form utility. In addition, another useful tool that will be added to the eduFuse system is the ability to determine which topics students have mastered and which topics students are having trouble mastering [Por98].

1.3 Components within eduFuse

There are four main components of the eduFuse system; these are the concept graph, editor, visualization, and collaboration. Each component in the system contains many different features that contribute to the system's overall goals of making the pedagogical process more efficient. The user interface in the eduFuse system has recently been redesigned to reflect the importance of navigation among these four components; these user interface changes will be discussed later in the paper.

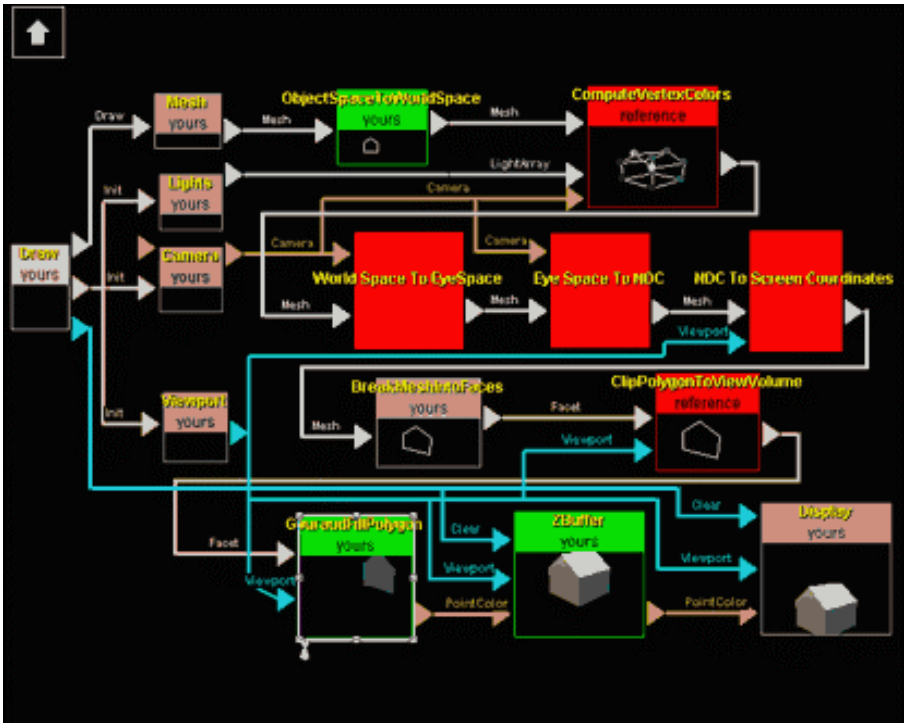

Fig. 1 – one example of a concept graph in the eduFuse system

The concept graph is the first component presented to users after they log in to eduFuse; it is intended to give users a top-level view of a programming assignment. Each concept graph represents a pipeline that consists of modules and connectors among those modules. In the eduFuse system, each module encapsulates either an algorithmic concept to be taught to the student, or auxiliary code necessary for the pipeline to function correctly. Each module contains a viewport, which is an area in which visual output of each algorithm is displayed. The connectors are data links that are color-coded according to the data type; these connectors provide input and output data to each module [Boy99]. By utilizing these visual tools, the concept graph provides a representation of a process that enables the student to easily determine how various algorithms contained in the modules interact with each other.

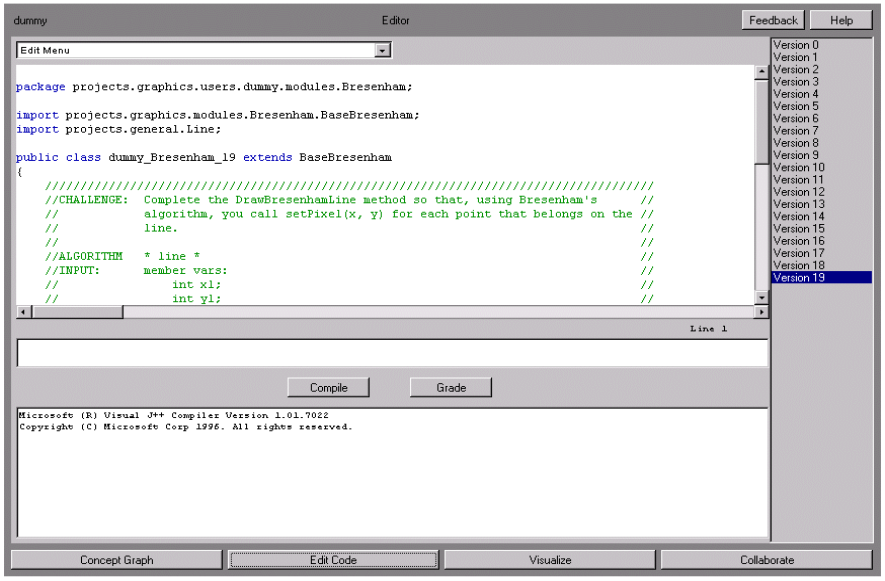

Fig. 2 – the code editor component in the eduFuse system

The code editor and visualization components are closely related since they are both operational only at the module level. These two components enable the student to examine a particular algorithmic concept in more detail. Within the editor component the student can view and edit

the code for a particular module using many of the features found in an editor contained in a professional development environment. In addition to the editor itself, this component contains a button for activating compilation, which is accomplished by sending the Java code to the server; the server then returns compiled Java bytecode that is interpreted by the JVM contained in the web browser.

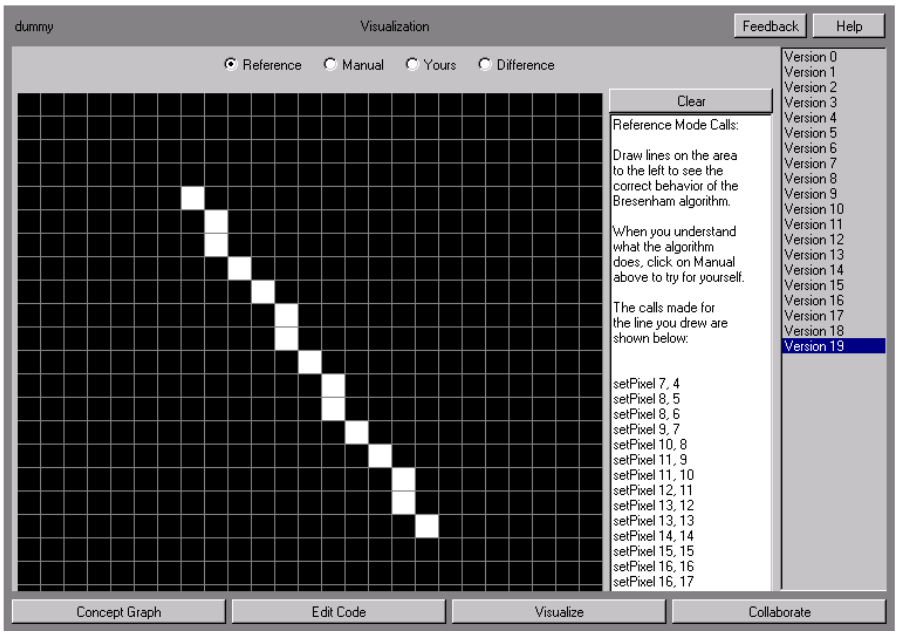

Fig. 3 – the visualization component in the eduFuse system

The visualization component is unique to the eduFuse system; it provides another visual representation similar to the concept graph. This component enables the student to easily determine the behavior of an algorithm by examining the returned outputs of specified inputs. A unique version history feature also ties together both this and the code editor components. Found along the right side of the editor and visualization components, the version history feature displays the successive copies of the student's code that has been saved every time it is

compiled. The version history is displayed in both the editor and visualization components so a student can choose to view successive versions in either component [Boy99].

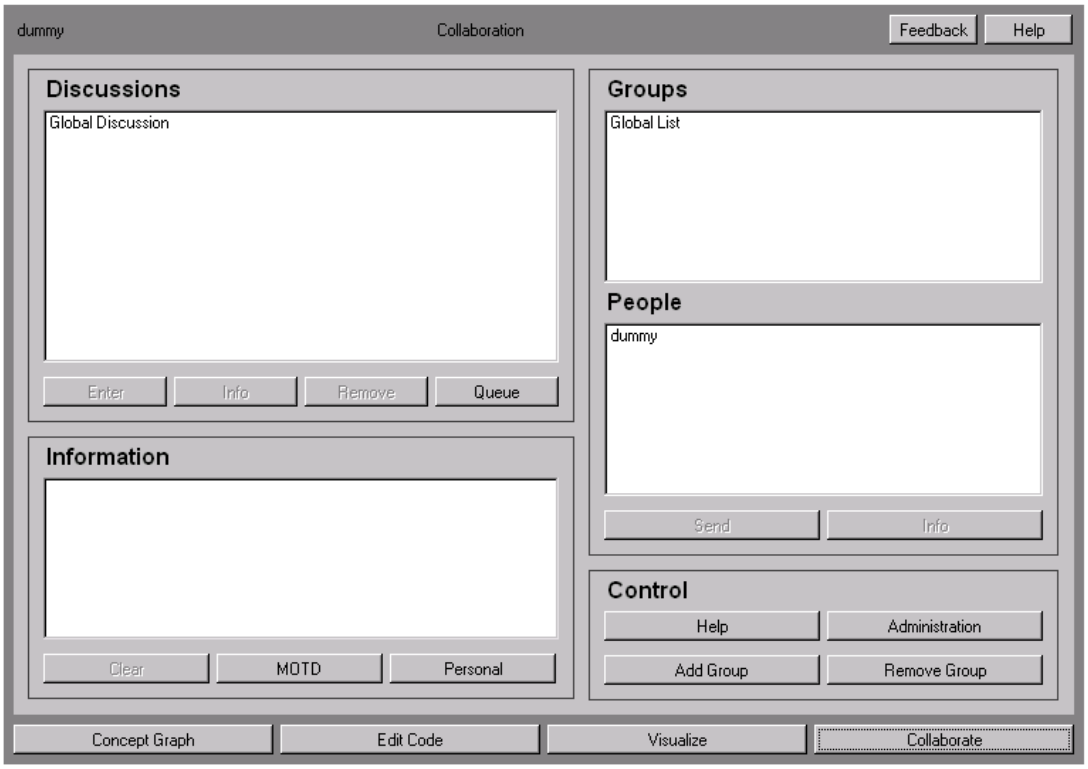

Fig. 4 – the collaboration component in the eduFuse system

The collaboration component contains several tools that are important for users to successfully communicate and collaborate with each other. The basic tools in the collaboration component include instant messaging and discussion group functionalities, which enable users to engage in one-on-one or multiple participant online conversations. All messages are stored in a central repository so the contents of a conversation can be retrieved at a later time. Along with these standard communication tools, eduFuse has a unique feature called the help queue. This feature provides functionality for students to submit questions in a queue format; after questions have been submitted, a teaching assistant or professor can go through the queue and answer questions

in the order students submitted them [Ken99]. The help queue feature provides an organized and fair method of distributing students' questions to a teaching assistant or professor.

1.4 History of eduFuse

The eduFuse system has been an idea since before 1996; during that year the system finally came into existence. Since that time, eduFuse has been a dynamic system, undergoing continual changes and improvements. Some of these changes were architectural, such as rewriting portions of the system from Java to Perl [Boy97]. Other changes have been more to enhance the usability of the system, such as those described below. In addition to these numerous changes to the system, eduFuse has also been deployed and used by university students at both MIT and Washington State University [Por98]. Valuable feedback from both students and instructors who used the eduFuse system was collected and used to guide future development.

2. Prior & current work on eduFuse

2.1 Module development

When I first joined the eduFuse project in January 1997, some of the first work I did was on module development. To become accustomed to the eduFuse system and its structure, I started by developing several modules that contained some elementary graphics algorithms. After doing this work, I discovered how simple it was to create modules in eduFuse and effectively design programming assignments for a university level class. The new modules that I created included implementations of z-buffer and polygon shading algorithms. The first, a z-buffer algorithm,

captures the points of a 3-dimensional image that are closest to a camera eye. The second, a polygon shading algorithm, attempts to shade a polygon with a specified starting color.

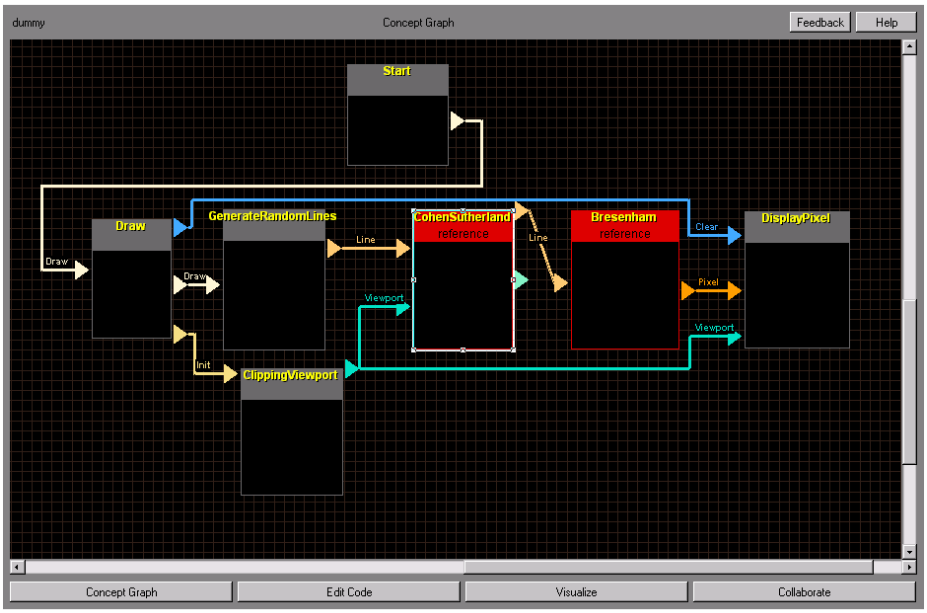

Fig. 5 – the new user interface for the eduFuse system

2.2 User interface redesign

Recently, with the addition of the collaboration component to the eduFuse system, I redesigned the user interface (UI) to reflect the importance of easy navigation among the four different components of eduFuse.

Fig. 6 – the locator buttons utilized by the new user interface

This new user interface aims to enable navigation among the different components of the system, along with providing contextual information to the user about various features of eduFuse. One

of the new tools of this redesigned UI is a set of four locator buttons along the bottom of the screen that can be used to jump between the four main components. The middle buttons, Edit Code & Visualize are only active when a module is selected since these two components are only operational on the module level.

Fig. 7 – the contextual information toolbar utilized by the new user interface

Near the top of the screen, the new user interface utilizes a toolbar that contains the user's name on the upper left corner, the user's location within the system in the middle, and two buttons on the upper right corner. The two buttons in the upper right are used for accessing help files on the system and for submitting feedback reports about the system to the eduFuse developers. The locator buttons along the bottom and the toolbar along the top are visible to the user at all times regardless of the component in which the user is working.

2.3 Help files feature

During the summer of 1999 eduFuse gained some features that were aimed at packaging the system better, to make it more usable by new users. One important feature that was added during this time is the help files functionality. Access to the help files feature is easily enabled with the use of the help button; this button, which is located in the toolbar, can be found in

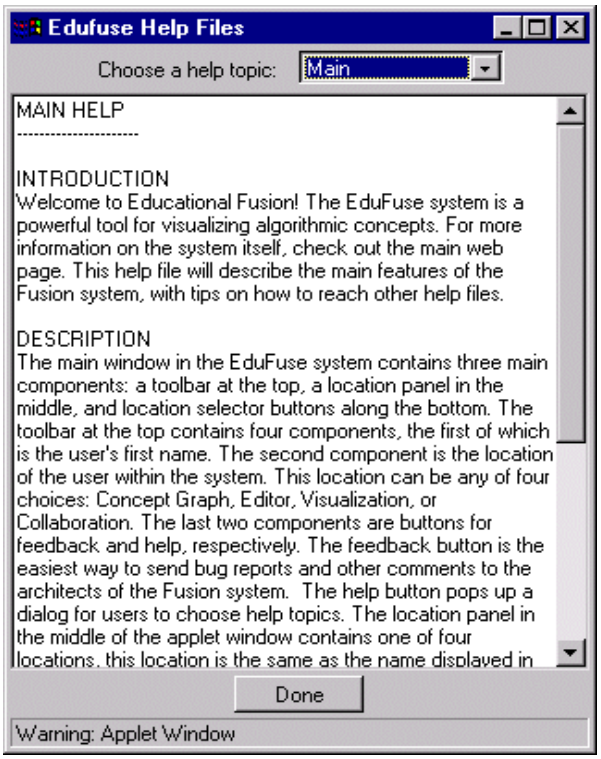

Fig. 8 – the help files feature in eduFuse

the upper right corner of every screen within the eduFuse system. After a user presses the help button, a dialog window pops up with which the user can easily access context-specific help with the use of a pulldown menu that lists the various help topics.

2.4 Grading forms feature

Most recently, a feature was added to eduFuse that directly aids in the goal of making instructors' time more efficient. This feature is the online grading forms utility, which enables instructors to view and grade students' submitted programming assignments. After an instructor views a student's code and submits a grade and comments, this information is stored in a pre-specified location within the system. The next time a student whose code has been graded logs in to eduFuse, he or she can check for new grading comments that may be available in the pre-specified location. The student can then easily view the grade and comments submitted by the instructor.

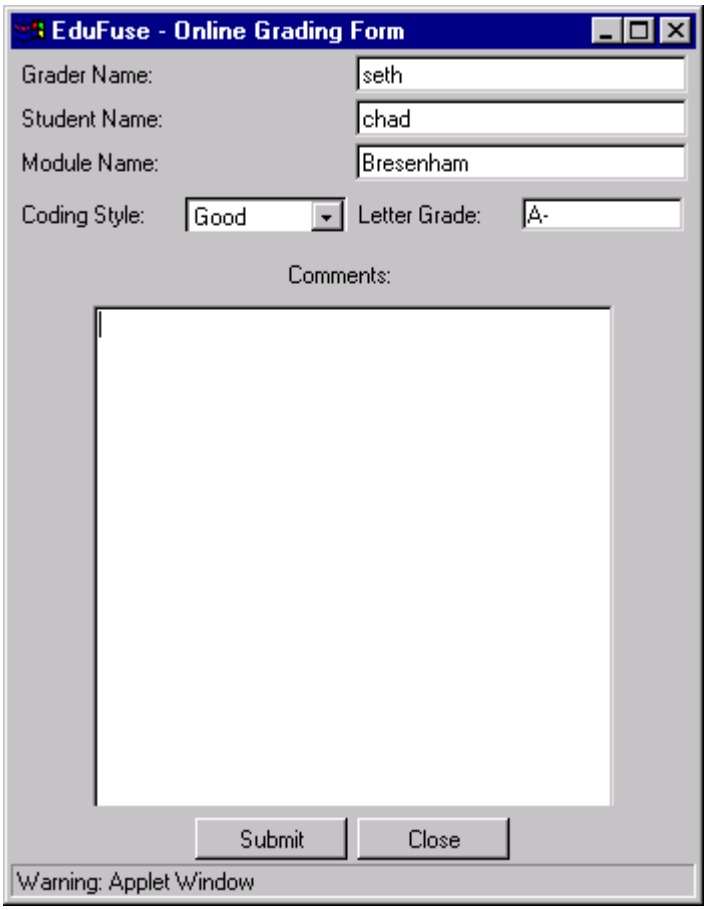

Fig. 9 – the online grading form feature in eduFuse

3. Benefits of changes

3.1 Simple module creation procedure

Developing a simple procedure to create a module is an important and unique feature of eduFuse, since each module contains an algorithmic concept that can become a programming assignment in a university-level class. Since the creation of assignments in a computer science class takes quite a bit of instructors' time, the eduFuse system achieves part of the goal of making instructors' time more efficient by making the module creation process simple. Instructors can spend less time on this task using the eduFuse system, which frees more of their time that can be spent interacting with students.

3.2 Improved navigability & usability

The main benefits of the redesigned user interface and the help files feature are the improved navigability among components in eduFuse, and the improved usability of the system. Using the locator buttons along the bottom of the screen, a user can easily jump to any component in the eduFuse system. In addition to this improved navigability, the toolbar along the top of the screen, containing various contextual information and the help and feedback buttons, have enhanced usability by making important information easy to find. The help files feature has also improved the overall usability of the system by providing clear, concise and thorough explanations of all the components within the eduFuse system; this feature is especially useful for first-time users.

3.3 Improved efficiency of pedagogical process

As stated previously, the overall goals of the eduFuse system are to improve the efficiency of the pedagogical process for both students and instructors. As detailed above, timesaving features across the eduFuse system help to achieve these goals. These features include the simple module creation process, the redesigned user interface, and the help files utility. One of the most recent features added to the system, the grading form utility, also aids in this goal of improving the efficiency of the pedagogical process. In one sense, it is also a timesaving feature similar to the others that makes an administrative task more efficient, thereby freeing instructors' time that can be spent interacting with students. The online grading form feature can also be a means for an instructor to better evaluate a student's submitted work, which can be very beneficial for determining a student's learning patterns. This information can then be used by instructors to design a curriculum that better serves the variety of learning styles found in any class.

4. Future work on eduFuse

4.1 Improvements on grading forms

There are many aspects of the online grading form utility that can be improved. For instance, pre-generated test input suites could be included in the feature; this improvement would involve the system automatically running a student's submitted code on a set of predetermined inputs and posting the results to the online grading form. With this improvement, an instructor would have a quick method for determining the correctness of the student's code; this addition of test input suites would make the grading process even more efficient. Another improvement to the online grading form feature could be a reporting functionality that would allow the instructor to

summarize one student's work for the term or the work of multiple students on a particular assignment. The instructor could utilize the results of these reports to design a better curriculum that reflects the abilities of the students in the class, or to determine if some students need extra assistance on various aspects of the course material.

4.2 Other tools for instructors in eduFuse

There are numerous other tools for instructors that could be incorporated into eduFuse. One such tool could be an online meeting utility, with which instructors could hold staff meetings to discuss curriculum, student work, or other class issues. This kind of tool would be especially useful in a true distance learning situation, where the instructor's teaching assistants or other colleagues are located in a different physical location. Another tool for instructors that has been discussed for addition to the eduFuse system is a desktop sharing mechanism. This type of tool would make the process of aiding students much easier by enabling an instructor or teaching assistant to control a student's desktop. With this type of control, the instructor could show the student specific mouse or keyboard actions that would appear on the student's desktop; this would be very helpful for many students. There are currently hooks in place for most of these features to be added to the eduFuse system.

5. Conclusion

The eduFuse system provides a robust online learning environment that contains numerous timesaving features within the system for both students and teachers that enable eduFuse to make the pedagogical process more efficient. These tools for students include the visual representation

provided by the concept graph and visualization components, the easy to use editor and the collaboration component that enables seamless communication with other users. For instructors, these timesaving tools include the simple module creation process and the online grading form utility, and will hopefully include other tools in the future. Also, all users can benefit from the redesigned user interface that enables quick navigation throughout the system, and the help files utility that provides context specific help with the use of a pulldown menu. Overall, the eduFuse system has at least partially achieved its goals of making the pedagogical process more efficient for both students and teachers; the system will hopefully undergo more improvements in the future that include the addition of more timesaving tools for both students and teachers.

Bibliography

[Boy97] Nathan D. T. Boyd. “A Platform for Distributed Learning and Teaching of Algorithmic Concepts.” MIT M.Eng. Thesis. 1997.

[Boy99] Aaron T. T. Boyd. “Educational Fusion: A Distributed Visual Environment for Teaching Algorithms.” MIT M.Eng. Thesis. 1999.

[Ken99] Kevin Kennedy. “Educational Fusion: Implementing a Production Quality User Interface with JFC.” MIT Advanced Undergraduate Project. 1999.

[Por98] Brandon W. Porter. “Educational Fusion: An Instructional, Web-based, Software Development Platform.” MIT M.Eng. Thesis. 1998.

[Tel94] Seth Teller. “NSF Career Development Plan.” MIT Department of Electrical Engineering and Computer Science. 1994.