Integrating a Community of Evolving Learners.

Jonah Benton and Elizabeth Sklar
Department of Computer Science
Columbia University
1214 Amsterdam Avenue, Mailcode 0401
New York, NY 10027 USA
{jrb2011,sklar@cs.columbia.edu}

Abstract

Columbia's Technology Integration Partnership (TIP) is developing a web-based software infrastructure to support interactive, real-time collaborative educational activities for a community of human learners and intelligent software agents. We support both cooperative and competitive activities, and have done initial deployment and testing of the environment in New York City public schools. Current and future work focuses on integration of these activities into existing TIP classroom activities in these schools; enhancing and adapting the software infrastructure for use by students, educators and developers outside the TIP project; and gathering data and incorporating additional pedagogical methods to better enable, measure and support human and agent learning over time.

Introduction

Since the introduction of personal computers in 1977, the market for educational software has grown very quickly. Today, most schools have computers on-site or programs for providing students with computers, and many have been "wired", giving teachers and students direct access to the Internet. Content management and administrative applications allow students to retrieve and submit their homework online and for parents to be in touch with teachers online. However, interactive, real-time educational applications are not in widespread use in real, non-laboratory classrooms.

Goals

We have been interested in creating a virtual community of learners, where participants engage in interactive, real-time multi-user activities in educational skill-oriented domains. Our goals are threefold: first, to use the Internet to create this kind of learning community, where learners in disparate locations can come together in real time and help each other advance. We seek to leverage the potential anonymity of Internet interactions to create learning partnerships between students based on their skill levels, rather than their ages or common locations, and to address fundamental concerns of safety and privacy.

Second, we seek to build comprehensive models of student abilities in particular learning domains, by gathering performance data about their activities. We can use these models to improve the learning process, by providing a picture to teachers of a student's strengths and weaknesses; by computationally determining appropriate pairings of students for multi-user activities, performing these matchings in real-time, while they are using the system; and by selecting appropriate individualized problems for each student, based on his/her past interactions with the system. We also hope to use these learning models to build intelligent agents that can serve as smart playmates, to overcome the potential real-time unavailability of human learning partners.

Third, we hope that in an educational climate increasingly focused on skills-oriented testing, our studies of these kinds of real-time interactive play/learning activities will contribute to the process of more appropriately tuning educational software to the needs of students. In addition, a system that has the ability to provide comprehensive reporting to teachers on their students' activities and performance while engaged in the system offers teachers another metric for measuring student achievement beyond standardized testing, perhaps a more realistic metric which is measured over time, in a non-stressful, everyday classroom environment.

The CEL System

Our project is an updated implementation of the previously developed framework called CEL (Community of Evolving Learners) [Sklar,2000]. CEL is an Internet-based system where students engage in two-player educational games. These games are straightforward (i.e., there is no glitz) and provide practice at basic skills (e.g., spelling, typing, math, geography). Players come to the CEL website, select a unique anonymizing graphical ID (called an "IDsign"), and enter the "playground" associated with each game. Once in a playground, a player can invite or be invited by another player to play a particular game. Once invited, a player can accept or refuse an invitation to play. When one player invites and another player accepts, both players are taken to the "game", usually implemented as an interactive applet.

The applets provide controls for each player to make moves in the game, and feedback so that they can see the results both of their moves, and of the other player's moves, in real time. Some of the games are competitive, where a player tries to do better than the other at particular skill-oriented tasks. Other games are cooperative, in which players try to improve on each others' moves to achieve a better "team" score.

When a game is finished, the moves and the performance of both players are stored and compared to their past performances at the same game as well as performances of other players. As below, this performance data is used to train software agents as playmates residing in the system. Each human player can then return to the playground to select another game/playmate to play or invite the same playmate to play again.

We support both fully anonymized play, as well as only externally anonymized play. In the former, players are identified only through graphical identifiers, have no opportunity for interaction or communication other than through moves in the game, and have no history or model data accumulated or stored about them. In the externally anonymized model, players must log in with their assigned username prior to entering the play area. Once in the play area, players are still anonymized through graphical identifiers, though the CEL system internally tracks their real identity and game performances over time. (Players receive accounts and have their performance data measured and tracked over time only with their parents consent.)

In either play model, move data generated by human players is also stored and used as training data for autonomous software agents, which can learn from this data and serve as playmates for players when other human players are not available. In a school setting, the performance data for a particular player over time will be known to the agent playmates, so that agents can adapt their play appropriately to the skill level of the player.

Our Software

Our system includes client and server software, written in Java and PHP. Our client software consists of a communications framework, written in Java which supports asynchronous, low-latency communications, on top of which interactive, move-oriented games can be developed (either synchronous, i.e., turn-taking, or asynchronous, where both players can make moves at the same time). The communications protocol is similar to that used in chat systems, and indeed, we may in the future adopt one of the open source communications toolkits, like Jabber.

Our server software is written in Java and PHP. The communications server that relays moves from player to player is written in Java. The community software that tracks players through playgrounds and their invite/accept interactions is written in PHP. The intelligent agents that serve as automated playmates are written in Java. Performance data is stored in a combination of log files and mySql database tables.

Role of CEL in the TIP project

Columbia's TIP (Technology Integration Partnership) project is a joint effort of Columbia University's School of Engineering and Applied Science (SEAS) and Teachers College (TC). It has two major goals:

-to increase the number of technologically competent teachers, certified in mathematics and the sciences, available and willing to work in the urban school environment; and

-to deepen the content knowledge and technology skills of teachers who are already teaching in the public schools.

The TIP project has successfully been using probeware and educational robotics, through mobile equipment labs since Fall 2002 in over 35 New York City public school classrooms, ranging from grades 5 through 12. As part of this project, we have built a database of educational robotics training materials, curriculum and specific lesson plans. We hope to similarly deploy the CEL project and develop lesson and activity plans for both in- and out-of-classroom learning.

Future Work

In addition to integrating CEL into the TIP program, we hope to enhance the software for use by interested parties outside the TIP research group, and to use accumulated data to refine learning and teaching models and improve and refine our activities.

References

Sklar, Elizabeth. CEL: A Framework for Enabling an Internet Learning Community. PhD Thesis, Brandeis University, 2000.

Acknowledgements

This work was supported by NSF #REC-01-13317 ("Creating One to One Learning Opportunities across the Internet") and NSF #DGE-00-86390 ("GK12: Integrating New Media Technologies into Teacher Development").