

Give Students a Clue: An AI Course Project

David M. Hansen, Ph.D.
Associate Professor
Computer and Information Science
George Fox University
dhansen@georgefox.edu
<http://cs.georgefox.edu/DepartmentPages/Research/Glomus>

April 14, 2006

Goals

The primary goal of this work has been to develop a robust system of software for a course-project that can be deployed in a traditional course in Artificial Intelligence at smaller college and universities. We have used a course-project based on the game of Clue at George Fox University, but the project has been hampered by the inability of student-developed computer agents to compete against one another without students manually operating their software agent. Thus the goals of this work have been: (1) to implement a simple centralized server in software that would allow students to test their agents and and compete against one another, and (2) to do so in such a way as to make the entire project accessible to other teaching faculty by providing production-quality software and extensive documentation.

Research Setting

The work has been conducted by faculty and staff in the Computer and Information Science Department at George Fox University. Under the guidance of faculty, student interns have developed robust protocols for client-server communication and have implemented a server and game-playing clients that can be extended by students during the course project.

The project, and the associated software, is currently being used in the course *CSIS 400 — Artificial Intelligence*. This beta-test is being used to validate and debug the software we have developed as well as to develop curricular materials that cover the integration of this project into a course in AI.

Project Results

We have now successfully completed the first three phases of our project plan. In the spring of 2005 we completed our software design and developed the specification for the client-server communication protocol.

Over the summer months we implemented the beta-version of our software server. The server is comprised of approximately 4500 lines of Java in 14 Java classes. The server has been designed to be robust in the event of a variety of failures including illegal client actions and basic connectivity failures.

In addition, we have developed three simple clients, two written in Java, and a third written in Prolog. Each client consists of a few hundred lines of code that provides a working, but “dumb” agent. The Prolog client, for example, plays a simple but legal game and is designed to be easily enhanced by students who add rules to its knowledge-base to implement “intelligent” behavior. A similar client has been implemented in Java providing a similar template for students who might wish to use Java instead. A third client is a Java applet that presents an interactive human user with a graphical user interface for playing the game. This client is purely for demonstration purposes, allowing a user to connect to our server and compete against two simple computer clients spawned by the server. This applet is currently available on our project web site.

Our server, together with these clients, is available for download from our project web site.

Our project is well on its way to successful completion. However, we are behind schedule in the development of curricular materials and our project web site. This is due to the inability to hire a second student during the summer of 2005. Given our limited resources last summer, we focused primarily on the development of the software server and our web site remains largely undeveloped.

Impact

The results of this work (the server and client templates) have greatly enhanced the course project. Three notable enhancements include

1. The development of extensible templates in Prolog and Java that allowed students to quickly make progress without having to learn minute programming language and server-protocol details. Students were able to extend the Prolog client very quickly with little instruction in the language itself.
2. An always-available testbed. In the past, students had to play other students by manually using their agent to play a game of Clue. This was tedious and error-prone and gave little opportunity for students to test their agents before the end of the course.
3. High-speed testing that results in useful feedback. The limited number of games students were able to play in previous years made it difficult to

assess whether or not a strategy employed by their agent was actually of value. Student can now add a new strategy to their agent and play old vs. new over thousands of games in a matter of minutes to determine whether or not the new strategy has improved their player. This has been one of the most satisfying outcomes of this work as students have been encouraged to employ a methodical and testable approach to developing their agents.

We look forward to the opportunity to disseminate our work via appropriate conferences where we expect to find faculty eager to adopt this project and use our software.