Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

INTEGRATION OF COMPUTER SIMULATION AND VISUALIZATION RESEARCH INTO UNDERGRADUATE DEGREE PROGRAMS

T. Andrew Yang
Department of Computer Science
Indiana University of Pennsylvania
Indiana, PA 15705, U.S.A.

ABSTRACT

Faculty from several departments in the College of Natural Sciences and Mathematics at IUP have engaged in interdisciplinary projects involving the simulation and visualization of neural networks and material science research. The existing Computer Science degree programs at IUP, however, contain no courses in computer simulation. It has been felt by all the researchers involved in the projects that a degree program focusing on computer simulation is needed to, among its other missions, cultivate students with sufficient knowledge and skills to participate in the projects. This paper starts with an analysis of the knowledge and skills required of the students, followed by identification of existing and new courses that may be taken to acquire the knowledge and skills. The paper concludes with a proposal to establish a degree program in computer simulation and visualization, and an approach in integrating the research projects with the proposed degree program.

1 INTRODUCTION

The significance of scientific simulation in scientific inquiry has dramatically increased for the past decade. The following is an excerpt that very well illustrates this:

"… computer simulation has now joined theory and experimentation as a third path to scientific knowledge. Simulation plays an increasingly critical role in all areas of science and engineering. However, as the uses of simulation expand, the need for high performance computing of increasing power, flexibility, and utility grows proportionately." (Source: The NSF Advanced Computational Research program announcement, NSF 98-168)

Simulation and visualization systems are traditionally stand-alone computer applications. For the past three to four years, with the widespread use of Web and the advancement of related technology, more Web-based simulation and visualization systems have been developed (Davis, Chen and Brook 1998, Pidd, Oses and Brooks 1999). Many of these developments employ distributed object-computing architecture, such as CORBA, RMI, EJB, and autonomous agents, to enable the effective deployment of the application over the Web. At IUP, a Web based real-time visualization system has been developed to enable remote viewers to interactively and dynamically manipulate neural network models and data generated by the simulation experiments (Yang 2001, Yang, Cross and MacMaster 2001). The major advantage of moving a stand-alone application to the Web is the wide availability of Web browsers, which can be used as the front-end user interface to the application by remote users, who may potentially be located across the globe. The development of a Web-based system for neural network simulations takes the simulation and the generated data to the Web, and allows remote users to configure and observe the simulation in real time.

The development project, however, has experienced problems in identifying qualified student programmers/researchers with appropriate set of knowledge and skills to participate in the research projects. Currently at IUP there exist no Computer Science courses in the field of computer simulation, not to mention that most of the Computer Science students lack the application knowledge of neural networks or material sciences researches. Students majoring in natural sciences may be equipped with sufficient application knowledge, but they are usually short of the required skills in developing a serious simulation and visualization software system.

It is apparent that there exists a gap that needs to be bridged. In the rest of this paper, an introduction to one of our major projects is introduced, that is, the on-line manipulation of neural networks. The establishment of a college-wide minor program focusing on computer simulation and visualization is then proposed. The program may be taken by all interested students as a minor. The program is designed in such a way that students completing the minor will have sufficient knowledge and skills to be competent practitioner in the field of computer simulation and/or visualization. The paper concludes with a proposed research lab to integrate the degree program with the on-going research projects.

2 THE APPLICATION: ON-LINE MULTIMEDIA MANIPULATION OF NEURAL NETWORKS

While visual data analysis tools has been commonly used in scientific simulation, they are typically used as stand-alone applications; that is, the viewers of the visualization must be physically close to the computer running the visualization (e.g., in a classroom or in a lab). The recent advancement in Web-based technology has enabled the development of on-line simulation and visualization systems. The ongoing project of developing an on-line visualization system has been collaboration between a research project on a novel attractor neural network architecture, the self-trapping network (STN) (Pavloski and Karimi 1999, 2001), and a group of Computer Science faculty and students at IUP, where a Web based visualization system has been developed to enable remote viewers to interactively and dynamically manipulate neural network models and the generated data.

By linking a neural network simulation engine to a Web-based visualization system, the data generated from the simulations can be remotely viewed by the users (researchers, teachers, and students, et. al.), who may be geographically separate from each other. Furthermore, with interactive and dynamic configuration capability, the visualization server can get commands and data input from a remote user, who may request the simulation engine to reconfigure its underlying network architecture, by, for instance, changing the number of neurons in the network or modifying the configuration of the neurons.

A Web-based visualization system, such as the one described above, offers major advantages to the learning community. A teacher will be able to show an interactive and real-time scientific model to the students, whether in the classroom or in distant locations, by zooming in and out of the model and by rotating the model for different viewing perspectives. Because the visualization system is linked to the simulation engine, the teacher may show the students a dynamic pattern that changes over time, given new data feeds from the simulation engine. The neural network visualization system that is being developed at IUP, for instance, is capable of showing the firing and un-firing of individual neurons in the neural network. With the real-time visualization system in place, the viewers, whether researchers or students, will be able to view the underlying neural network going through the dynamic change in real time.

Furthermore, an integrated simulation and visualization system allows remote users to interact with the simulation engine, by, for instance, re-configuring the underlying network architecture and viewing the effect of change after re-running the experiment. In a distributed learning environment, this ability to remotely manipulate a virtual experiment will greatly increase the realness of the subjects that may involve experiments and demonstrations, such as in physics, chemistry, and biological sciences, et. al.

3 THE USAGE OF ON-LINE VISUALIZATION

In the current project, client server technologies such as Java applets (Yang Linn and Quadrato 1998) and sockets programming have been used to develop the on-line visualization system. The system is composed of a visualization server and a visualization client. Figure 1 illustrates the high-level design of a Web-based neural network simulation and visualization system by showing its three components (the neural network simulation program, the visualization server, and the Web browser) and their respective interactions. The neural network program is written in C/C++. The visualization server is a Java application, and the visualization client is a Java applet. The visualization client is originally stored on the Web server and is loaded by a Web browser when a user connects to the visualization server. Once loaded, the client opens sockets connections to the server, which then transmits the simulation result over the socket connection to the client. The client then displays the data as graphics within the Web browser.

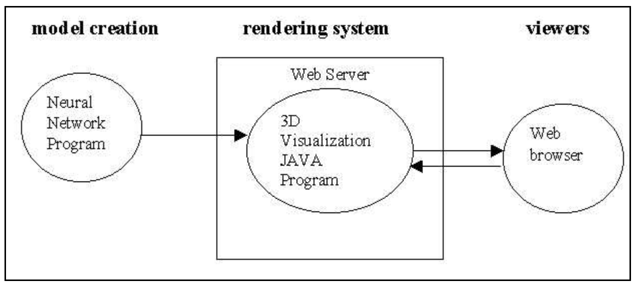

Figure 1: The Components in a Web-Based Neural Network Visualization System

A typical usage of this system involves the following steps:

- First of all, the viewer uses a Web browser to connect to the Web site where the visualization client is located.
- After a network connection is made, the user clicks the 'Load Network' button to have the selected network loaded into the visualization client, which then displays the neural network in the viewer's Web browser.
- The user then uses keystrokes and the mouse to view the 3D network.

Detailed discussion about the system and its usage is covered in a separate paper (Yang 2001).

4 ESSENTIAL KNOWLEDGE AND SKILLS FOR DEVELOPING WEB-BASED SIMULATION AND VISUALIZATION SYSTEMS

For a student programmer/researcher to successfully participate in research projects such as the one described above, the following knowledge and skills are considered essential:

- Computer programming:
  In addition to fundamental structured programming ability that is available from Computer Science courses such as *Programming One*, a student also needs knowledge and skills in *Object-Oriented Programming* and *Internet Programming*. Due to the particularly important role that the Java language plays in distributed computing, it is desirable that the student be a capable Java programmer.
- Mathematics:
  Effective programming for simulation and visualization applications demands sufficient Mathematical knowledge, which may be acquired from courses such as *Calculus* and *Linear Algebra*. A student may benefit from advanced course focusing on *Simulation Models*.
- Natural sciences:
  Due to the wide variety of subjects in Natural Sciences, it would be impractical to require a student to take a course from each of the subject areas. It is, however, important for a student to have some exposure to at least one fundamental scientific field, such as Physics. It is expected that the actual project in which a student participates would require knowledge in other scientific fields such as Chemistry or Geosciences. It is, however, a common practice in software development that a developer goes through the analysis stage to understand the application domain, before the rest of the development takes place. With fundamental scientific knowledge acquired from a natural science course, it is anticipated that the student would be able to pick up knowledge specifically needed for a new application.

It is apparent from the above analysis that an interdisciplinary program is needed to meet our objectives of cultivating student programmers/researchers who are capable of developing simulation and/or visualization programs or conducting research experiments using the developed tools. As illustrated in Figure 2, Simulation and Visualization (SAV) is proposed as a concentration incorporating specialty knowledge from the three underlying subject areas. An interdisciplinary minor is proposed to guide students to acquire knowledge and/or skills from the three areas and become qualified programmers and/or researchers in the simulation and visualization projects.

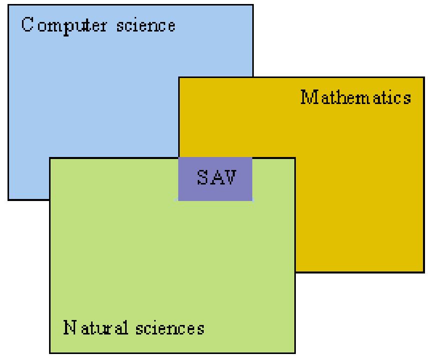

Figure 2: Simulation and Visualization (SAV) as an Inter-Disciplinary Area of Study

In the rest of the paper, requirements of the SAV minor program is covered, followed by further discussion of integrating research projects with undergraduate education via a research laboratory.

5 MINOR IN SCIENTIFIC SIMULATION AND VISUALIZATION

Table 1: Requirements for SAV Minor

Total number of credits: 18.
Core Courses: 9 credits
COSC304 Java Programming
MATH122 Calculus II
MATH371 Linear Algebra
+ At least one natural science course with lab, such as PHYS111/121 Physics I Lecture/lab. (Note: At IUP a natural science course with lab is part of the liberal studies requirements.)
Electives: 6 credits (Two out of the following courses)
COSC415 Internet Architecture and Programming
MATH447 Simulation Models
PHYS481 Computational Physics Applications
PHYS499 Natural Science Applications of Non-Linear Dynamics and Chaos
PSYC480 Introduction to Neural Networks
Capstone Course: 3 credits
COSC499 Scientific Simulation and Visualization

The Simulation and Visualization Minor is designed as a program to be offered by the College of Natural Sciences & Mathematics for students who are interested in acquiring specialty knowledge and skills in the field of scientific simulation and visualization. Detailed requirements for the minor are listed in Table 1 below.

Combining fundamental knowledge and research result from multiple disciplines, this minor is to offer students an essential training in computer programming, mathematics, and natural science, plus advanced electives on various areas related to scientific simulation and visualization. The total required 18 credits are distributed among three categories. The core allows the students to build strong foundation in object-oriented Internet programming and mathematics. The courses in the electives allow students to focus on Web-based development, computational sciences, and neural networks, et. al.

In addition to the core and elective courses, students are required to complete a capstone course (*Scientific Simulation and Visualization*), which provides them opportunities to integrate their knowledge and skills by participating in hands-on development projects and to interact with faculty researchers.

6 SIMULATION AND VISUALIZATION RESEARCH LABORATORY (SAVRL)

Due to its interdisciplinary nature of the SAV minor program, it is best administered and maintained by an interdisciplinary unit in the campus. A grant-funded research laboratory (SAVRL) is proposed to be in charge of the SAV program. The lab provides a focal point in the campus for faculty and students to engage in the study and development of theories, methodologies, and tools related to scientific simulation and visualization. SAVRL is staffed with faculty from multiple disciplines, especially those in the College of Natural Sciences & Mathematics. Two primary missions of SAVRL are teaching and research. By offering courses to support the college-wide Simulation and Visualization minor, SAVRL faculty are engaged in the creation and updating of courses related to scientific simulation and visualization, by, for example, integrating their research result into the curriculum. By hiring students minored in Simulation and Visualization to work in the lab, SAVRL provides a project-based research and development environment for students to further their knowledge and skills in this field.

7 SUMMARY

The gap existing between our needs of capable student researchers and developers in the area of computer simulation/visualization and the lack of such qualified students in our institution is examined, by first analyzing the needed knowledge and skills a qualified student must possess. An integrated approach consisting of a college-wide minor program (SAV) and a research lab (SAVRL) is then proposed to bridge the gap. The SAV minor provides students a concentrated area of study incorporating three areas: Computer Science, Mathematics, and Natural Sciences. The SAVRL serves as the administering unit of the SAV program and provides a repertoire of specialty knowledge, tools, and facilities to help faculty and students to carry out computer simulation and visualization research.

REFERENCES

Davis, Wayne J., Xu Chen, and Andrew Brook, 1998. Implementing on-line simulation upon the world-wide Web. *Proceedings of the 1998 Winter Simulation Conference*.

Pavloski, Raymond P. and Majid Karimi (1999). Control of basins of attraction in a self-trapping neural network with near-neighbor synapses. *Proceedings of the Third International Conference on Cognitive and Neural System*.

Pavloski, Raymond P. and Majid Karimi (2001). Self-trapping produces attractors in binary networks with sparse connectivity. Submitted to *Neural Networks*.

Pidd, Michael, N. Oses, and R.J. Brooks (1999). Component-Based Simulation on the Web? *Proceedings of the 1999 Winter Simulation Conference*.

Yang, T. Andrew, James Linn and David Quadrato (1998). Developing Integrated Web and Database Applications Using JAVA Applets and JDBC Drivers. *Proceedings of the 29th ACM SIGCSE (Special Interest Group in Computer Science Education) Technical Symposium*.

Yang, T. Andrew (2001). Development of a Distributed Visualization System for Neural Network Simulation. *Proceedings of the International Joint Conference on Neural Networks*.

Yang, T. Andrew, Zac Cross, and Scott MacMaster (2001). Development of a Web-based Visualization System for Scientific Simulation. *Proceedings of the IEEE International Conference on Multimedia and Expo.*

AUTHOR BIOGRAPHY

T. ANDREW YANG received his Ph.D. from University of Minnesota in 1993, and has since been working as a Computer Science educator and researcher. He is currently an associate professor of Computer Science at Indiana University of Pennsylvania, Indiana, PA. His research interests include Distributed Systems Development, Computer Simulation and Visualization, Computer Science Education, and Secure Software Development. He is a member of IEEE, SCS, and ACM. He can be reached by email at <yang.andrew.t@acm.org>.