Executive Summary on PetaScale Education Project
D. E. Stevenson
Director, Institute for Modeling and Simulation Applications
Clemson University

My experiences in computational science education now span almost two decades and one thing has been constant: students and faculty alike are unable to formulate models and construct simulations that are not only efficient but verifiable and validatable. As a member of the board of directors of Shodor Education Foundation and a co-PI on the National Computational Science Institute, it has been clear that while problem-solving environments are efficient teaching tools, a large gap exists between such tools and effective use of supercomputing resources. As a member of the Technical Working Group on Verification, Validation, and Accreditation at the Defense Modeling and Simulation Office, I have worked to develop guidelines for averting gross misjudgments that come from unjustified simulations. That experience has been put into the CSERD Pathways project of the National Science Digital Library.

The project that I propose is a “K through Gray” education program based on introductory problem-solving environments developed from a toolbox of components to be developed specifically to explore common modeling and simulation paradigms. The education program must develop subjects in a manner suitable for computational treatment, using modeling metaphors used by subject matter experts. Obvious treatments are deterministic, non-deterministic, stochastic, and uncertain dynamical systems, agent-based systems, and alternative systems such as fuzzy and probabilistic logics and arithmetics. I predict that the peta-scale initiatives will start with conventional applications in STEM subjects, but U. S. competitiveness issues and world-wide problems in the life sciences, economics, and global climate will require computational science to educate all disciplines — not just the convention STEM disciplines — in computational techniques.

It is time researchers stopped writing their own codes; I know no serious scientist or engineer who considers programming their own codes their central focus but are required to do so because of lack of support. The fact that we continue to use the original 1960s programming paradigm is ludicrous; a body of evidence exists, culminating in the AAAS and the PITAC reports on supercomputing effectiveness being blocked by the inability to develop software.

The problem with many of the available environments is their lack of flexibility. Real world problems do not fit into one problem-solving technique. Development of problem-solving environments must be supported by common tools based on the priorities set by the educational and practical needs of the scientists. The object-oriented paradigm is a step in this direction, but is still too low a level to be of practical value to working scientists focusing on modeling — and solving — 21st Century issues.

In effect, such a toolbox replaces the low-level programming paradigms of today by

• Simplifying development of visual interfaces for new and novel environments.
• Providing symbolic manipulation capabilities to pass the model on to simulation.
• Simplifying visualization support.
• Providing verification and validation for the models and verification of the simulations.
• Providing a multitude of simulation paradigms.

The toolbox must be supported with a kernel that can both communicate with middleware and provide advanced functioning. In order to develop verified, validated simulations, the kernel must provide more services than a library of solution mechanisms; it must provide for the mathematical development of new models that can be trusted.

315 McAdams Hall, Department of Computer Science, Clemson University, PO Box 341906, Clemson, SC 29634-1906
E-mail address: steve@cs.clemson.edu