

A Community Approach to Earth Systems Modeling

PAGES 117–118

Earth science often deals with complex systems spanning multiple disciplines. These systems are best described by integrated models built with contributions from specialists of many backgrounds. But building integrated models can be difficult; modular and hierarchical approaches help to manage the increasing complexity of these modeling systems, but there is a need for framework and integration methods and standards to support modularity. Complex models require many data and generate lots of output, so software and standards are required for data handling, model output, data distribution services, and user interfaces. Complex modeling systems must be efficient to be useful, so they require contributions by software engineers to ensure efficient architectures, accurate numerics, and implementation on fast computers. Further, integrated model systems can be difficult to learn and use unless adequate documentation, training, and support are provided.

Meeting all of these requirements can exceed the resources of typical research teams, and even those of a government agency, so there is a clear need for good mechanisms for designing, building, testing, and maintaining complex modeling systems. One such mechanism is the community modeling approach.

A community modeling system is an open-source (OS) suite of modeling components coupled in a framework. The system emerges through the collective efforts of a community of individuals who code, debug, test, document, run, and apply the modeling system. The community often includes both developers and users and may be distributed among different institutions and organizations.

Community models first emerged in the Earth sciences in the 1980s as a means to address the challenge of developing and applying complex models in the fields of air

quality modeling, climate prediction, and weather forecasting. Since then, an increasing number of community modeling projects have emerged. This article highlights specific strategies that reflect the promise and challenges of community modeling in Earth and environmental sciences.

An Open-Source, Community Approach for Complex Modeling Systems

An increasing number of community modeling projects have emerged over the past 3 decades. The first generation of community models, including the U.S. Environmental Protection Agency (EPA) Models-3 System (Community Multiscale Air Quality modeling system (CMAQ); http://www.epa.gov/asmdnerl/CMAQ/cmaq_model.html) [Byun and Schere, 2006], the National Center for Atmospheric Research (NCAR) Community Climate Model (CCM; <http://www.cgd.ucar.edu/cms/ccm3/history.shtml>), and the Pennsylvania State/NCAR Mesoscale Model (MM5; <http://www.mmm.ucar.edu/mm5/overview.html>), demonstrated that freely available, portable, well-documented, OS models would be enthusiastically received and used by the broader scientific community as research tools.

The next generation of community modeling projects was more ambitious. The Community Climate System Model (CCSM; <http://www.cesm.ucar.edu/models/atm-cam/>), the successor to CCM, continues to incorporate new physical processes and even human impacts at an accelerating rate. The CCSM project participated in the demanding Intergovernmental Panel on Climate Change assessments while continuing to serve as a vehicle for research. The Weather Research and Forecast (WRF) model (<http://www.wrf-model.org/index.php>), the successor to MM5, has attempted to serve both the research and operational communities. These models are widely used and have developed networks of contributors. They have also struggled to meet the demands placed on them: to satisfy diverse user bases, to keep up with the integration of new science, and to create governance bodies that can support scientific processes and scale to large numbers of participants.

More recently, much attention has been given to integrated modeling, which brings together different models from various disciplines to work together through exchanging data and information within the same framework [Argent, 2004; Gaber et al., 2008]. It is in this context that researchers in integrated environmental modeling and related domains such as Earth surface dynamics, hydrology, and some geographically focused areas (e.g., the Chesapeake Bay; see Figure 1) are seeking to organize and create new community modeling systems.

Some examples of integrated modeling projects in Earth science include the U.S. National Science Foundation (NSF)-funded Community Surface Dynamics Modeling System (CSDMS; <http://csdms.colorado.edu>); the EPA-funded Community Modeling and Analysis System (CMAS; <http://www.cmascenter.org/>); the U.S. National Oceanic and Atmospheric Administration (NOAA)-funded Chesapeake Community Modeling Program (CCMP; <http://ches.communitymodeling.org>); the Community Sediment-Transport Model System (CSTMS; <http://cstms.org>), supported through the National Oceanographic Partnership Program; and others. Yet another effort has been instigated by EPA, the Community for Integrated Environmental Modeling, also known as the Integrated Modeling for the Environment (IM4E) effort (<http://groups.google.com/group/commiem?hl=en>). These initiatives are less focused on individual processes and are more about arranging and linking various model components in a flexible and transparent way. Key to these efforts is a culture of scientific research based on collaborative development and open sharing of information and skills [Maxwell, 2006]. In contrast to the previous community models, here the communities are formed around more general topics and research areas and are not centered on a particular model or modeling system.

Advantages to a Community Approach

There are several advantages to a community approach [Voinov et al., 2008]. It provides a way to integrate effort among multiple institutions, which is crucial because Earth systems models are too multidisciplinary and complex for individual research groups. Community engagement can maintain project momentum and more project robustness in the face of uncertain funding and institutional support. An open,



Fig. 1. The migration of turbid floodwaters down the Chesapeake Bay after the severe floods of 2004 (true-color image captured by Aqua Moderate Resolution Imaging Spectroradiometer (MODIS) on 26 June 2004). The Chesapeake Bay is the focus region for the application of community models such as the Weather Research and Forecast (WRF) model and integrated modeling projects such as the Community Surface Dynamics Modeling System (CSDMS) and the Chesapeake Community Modeling Program (CCMP).

community approach can decrease redundant efforts because new models can be built upon already existing concepts, algorithms, and code.

Additionally, community modeling systems are often closely linked with their users, which promotes user participation and input at early stages of the project and during the testing phase. More user input allows for wider and more diverse testing, more robust models, and wider understanding and acceptance of results.

Most community modeling efforts rely on OS code. OS and its philosophy [Jesiek, 2003] satisfy the practical need of allowing many developers access to examine

and modify the code. There is significant experience in protecting intellectual property rights gained in OS, as well as in open-data communities. Organizations such as the Open Geospatial Consortium, Inc. (OGC; <http://www.opengeospatial.org/>), have developed a variety of licensing schemes, which can be well applied to models. Moreover, OS provides complete information transfer, and this transparency is important because code is the ultimate statement of the scientific understanding embodied in a numerical model. OS also facilitates peer review and replication of results, and it can be more easily reused, helping to reduce redundancy. Finally, OS seems appropriate for publicly

funded science projects because it ensures delivery of the results to the public.

Challenges

Complex systems are inherently hard to build and maintain, regardless of the approach, so building Earth systems models will never be easy. Researchers and administrators are still learning how best to develop OS scientific software using a community approach. There are technical challenges, including the need to develop fundamental algorithms to describe processes and implement these in efficient code. All of the other aspects of the model system

must be designed, integrated, and built, including software for manipulating, analyzing, and assimilating observations and to facilitate collaborations; standards and ontologies for data and model interfaces; and substantial improvements in hardware (e.g., network and computing infrastructure) [Hill et al., 2004; Kumfert et al., 2006; Moore and Tindall, 2005; Collins et al., 2005; Raymond, 2000].

However, the most difficult challenges can often be social or institutional. In many institutions the scientific reward structure is skewed toward publications and away from technical contributions. Funding is discontinuous and not reliably available for long-term support of technical infrastructure. Intellectual property policies of universities and private companies may be incompatible. Software is often viewed as a competitive advantage among competitors for funding and academic honors.

There are inefficiencies associated with informal project organizations that lack hierarchical structure. Many community projects are organized like bazaars [Raymond, 2000], with simultaneous efforts by many participants and without clear management, subordination, responsibilities, or strategies to deal with conflict and inefficiency. Informal management is not conducive to deadlines or customer-driven deliverables. It is also often difficult to work across disciplines, distances, and time zones with a diverse group of people, and to communicate effectively among scientists, engineers, users, and decision makers, who may have their own culture, vocabulary, and objectives.

What Is Needed?

Suggestions for supporting community modeling efforts and enhancing their success generally fall into two categories: organizational and technical. The organizational suggestions address the cultural and social background that is important for community modeling, as well as the programmatic decisions that can make projects more successful. The technical suggestions concern the actual software and analytical tools that are required. Within this framework, suggestions can be tailored to specific segments of the Earth science modeling community.

Funding agents and program managers should require that code be OS and meet a minimum level of standards or protocols as a prerequisite for receiving public funds. They should recognize the value of stable (longer-term) funding of software architects and engineers within the research environment, on par with the technical staff support of large academic or medical labs. They should support repositories of models and software and

ensure that researchers exchange information and standards among themselves. Code and documentation should be accessible as early and openly as possible during development to ensure that code from completed projects is archived and accessible, in the same way that field data and measurements are now. Model output from experiments should be made available to assist model validation and evaluation.

Further, institutional leadership should recognize the value of producing OS code and contributing to community modeling efforts to support collaborative environments that minimize the need for temporal and spatial localization. Producing well-documented, peer-reviewed code should become worthy of merit, while effective ways of peer review, publication, and citation of code, standards, and documentation should be introduced. OS should be embraced as a means of protecting intellectual property rights.

Community modeling project leaders should also encourage communication between scientists, technicians, and end users and should develop realistic criteria and metrics for success, considering project objectives, scope, and resources. Project governance should be formalized and enable teams to set priorities and make decisions as a unified effort working toward a common goal. Project governance must accommodate, and also be able to supersede, the interests and priorities of individuals, subgroups, disciplines, or institutions participating in the project.

Developers and the broader modeling community should adopt existing standards for data, model input and output, and interfaces. They should also help to develop standards for model conceptualization, formalization, and scaling. A good strategy may be to understand, use, and adapt existing tools first before developing new ones. However, if new tools are needed, those involved should provide good documentation, including examples and test cases.

Good software development practices should favor transparency, portability, and reusability and should include procedures for version control, bug tracking, regression testing, and release maintenance.

Making the Complex Easier

There are significant scientific and technical challenges associated with constructing complex Earth systems models. Overcoming these difficulties will require a collaborative modeling approach based on the fundamental principles of open scientific research, including sharing of ideas, data, and software. Improved software design and systems architecture in support of distributed community modeling efforts could significantly

increase the efficiency and utility of the community approach.

However, it is unlikely that the technical problems can be resolved unless the cultural problems of community modeling can be resolved. Thus, concerted progress toward more efficient community modeling will require the efforts of participants at all levels.

References

- Argent, R. M. (2004), An overview of model integration for environmental applications—Components, frameworks and semantics, *Environ. Modell. Software*, 19(3), 219–234.
- Byun, D., and K. L. Schere (2006), Review of the governing equations, computational algorithms, and other components of the Models-3 Community Multiscale Air Quality (CMAQ) modeling system, *Appl. Mech. Rev.*, 59(2), 51–77.
- Collins, N., G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva (2005), Design and implementation of components in the Earth System Modeling Framework, *Int. J. High Performance Comput. Appl.*, 19(3), 341–350.
- Gaber, N., G. Laniak, and L. Linker (2008), Integrated modeling for integrated environmental decision making, *White Pap.*, 100/R-08/010, 69 pp., U.S. Environ. Prot. Agency, Washington, D. C.
- Hill, C., C. DeLuca, V. Balaji, M. Suarez, and A. da Silva (2004), The architecture of the Earth System Modeling Framework, *Comput. Sci. Eng.*, 6(1), 18–28.
- Jesiek, B. K. (2003), Democratizing software: Open source, the hacker ethic, and beyond, *First Monday*, 8(10).
- Kumfert, G., D. E. Bernholdt, T. Epperly, J. Kohl, L. C. McInnes, S. Parker, and J. Ray (2006), How the Common Component Architecture advances computational science, *J. Phys. Conf. Ser.*, 46, 479–493.
- Maxwell, E. (2006), Open standards, open source, and open innovation: Harnessing the benefits of openness, Comm. for Econ. Dev., Washington, D. C.
- Moore, R. V., and C. I. Tindall (2005), An overview of the open modelling interface and environment (the OpenMI), *Environ. Sci. Policy*, 8(3), 279–286.
- Raymond, E. S. (2000), *The Cathedral and the Bazaar*, O'Reilly, Sebastopol, Calif.
- Voinov, A., R. R. Hood, J. D. Daus, H. Assaf, and R. Stewart (2008), Building a community modeling and information sharing culture, in *State-of-the-Art and Futures in Environmental Modelling and Software*, edited by A. J. Jakeman et al., pp. 345–365, Elsevier, New York.

Author Information

Alexey A. Voinov, International Institute for Geo-Information Science and Earth Observation, Enschede, Netherlands; Cecelia DeLuca, National Center for Atmospheric Research, Boulder, Colo.; Raleigh R. Hood, Center for Environmental Science, University of Maryland, Cambridge; E-mail: rhood@umces.edu; Scott Peckham, CSDMS Integration Facility, University of Colorado, Boulder; Christopher R. Sherwood, Coastal and Marine Geology, U.S. Geological Survey, Woods Hole, Mass.; and James P. M. Syvitski, CSDMS Integration Facility