Integrated Environmental and Earth Systems Modeling: a Community Approach

Alexey Voinov
Chesapeake Research Consortium
aavoinov@gmail.com

Chesapeake Community Modeling program

Chesapeake Community Modeling program

- The CRC Chesapeake Community Modeling program (CCMP) has been implemented through the decisions of the June, 2002 meeting of the CRC member institution modeling community and the subsequent Board approval in July, 2003

Chesapeake Community Modeling program

- The CRC Chesapeake Community Modeling program (CCMP) has been implemented through the decisions of the June, 2002 meeting of the CRC member institution modeling community and the subsequent Board approval in July, 2003
- The goal of the CCMP is to develop a community modeling system of model code, databases, and documentation for all aspects of basin processes. Model code will be distributed over the world-wide-web for use and revision

Chesapeake Community Modeling program

- The CRC Chesapeake Community Modeling program (CCMP) has been implemented through the decisions of the June, 2002 meeting of the CRC member institution modeling community and the subsequent Board approval in July, 2003
- The goal of the CCMP is to develop a community modeling system of model code, databases, and documentation for all aspects of basin processes. Model code will be distributed over the world-wide-web for use and revision
- Operational funding provided by NOAA

Chesapeake Community Modeling program

Linden Group charge:
- Models should be open source and supported by a substantial user community
- Models should have institutional homes.
- Data integration, prediction and uncertainty quantification are essential aspects of the modeling process.
- Modeling activity should be integrated into the educational mission of the CRC institutions.
- Models should be incorporate modern numerics as well as physical/biological parameterizations.

Chesapeake Community Modeling Program

A CCMP project

chesapeake community modeling program

"advancing the cause of accessible, open-source environmental models in support of research & management efforts"

CCMP Home | About | News | Wiki | Models & Data | Proposals & Funding | Workshops | Links

CCMP Home » Models & Data » Chesapeake Bay ROMS Community Model

CHESAPEAKE BAY ROMS COMMUNITY MODEL

Welcome to CCMP's ChesROMS homepage. This page will introduce you to the Chesapeake Bay ROMS Community Model (ChesROMS) as well as provide links to additional information and resources.

1. Introduction
2. Project Summary
3. Investigators
4. ChesROMS on Sourceforge

Introduction

ChesROMS is a community ocean modeling system for the Chesapeake Bay region being developed by scientists in NOAA, University of Maryland, CRC (Chesapeake Research Consortium) and MD DNR (Maryland Department of Natural Resources) supported by the NOAA MERHAB program. The model is built based on the Rutgers Regional Ocean Modeling System (ROMS, http://www.myroms.org/) with significant adaptations for the Chesapeake Bay.

The model is developed to provide a community modeling system for nowcast and forecast of 3D hydrodynamic circulation, temperature and salinity, sediment transport, biogeochemical and ecosystem states with applications to ecosystem and human health in the bay. Model validation is based on bay wide satellite remote sensing, real-time in situ measurements and historical data provided by Chesapeake Bay Program.

back to top

Project Summary

Various noxious and toxic algal blooms afflict the Chesapeake Bay and other coastal U.S. waters, posing threats to human health and natural resources. The goal of this regional study is to develop and implement an operational system that will nowcast and forecast the likelihood of blooms of the following three harmful algal bloom (HAB) species in Chesapeake Bay and its tidal tributaries: the dinoflagellates Karlodinium micrum and Prorocentrum minimum and the cyanobacteria Microcystis aeruginosa. In addition, the feasibility of predicting other HAB species will be investigated and pursued. The method proposed involves using real-time and 3-day forecast data acquired and derived from a variety of sources and techniques to drive multi-variate empirical habitat models that predict the probability of blooms caused by these particular HAB species. The predictions, in the form of maps, will be available via the World Wide Web to individuals and interested agencies to guide research, recreational and management activities. In particular, these nowcasts and forecasts will be employed

ChesROMS Introduction

General ChesROMS Info
- ChesROMS Home
- Introduction
- Project Summary
- Investigators
- Sourceforge ChesROMS Page

ChesROMS on Sourceforge

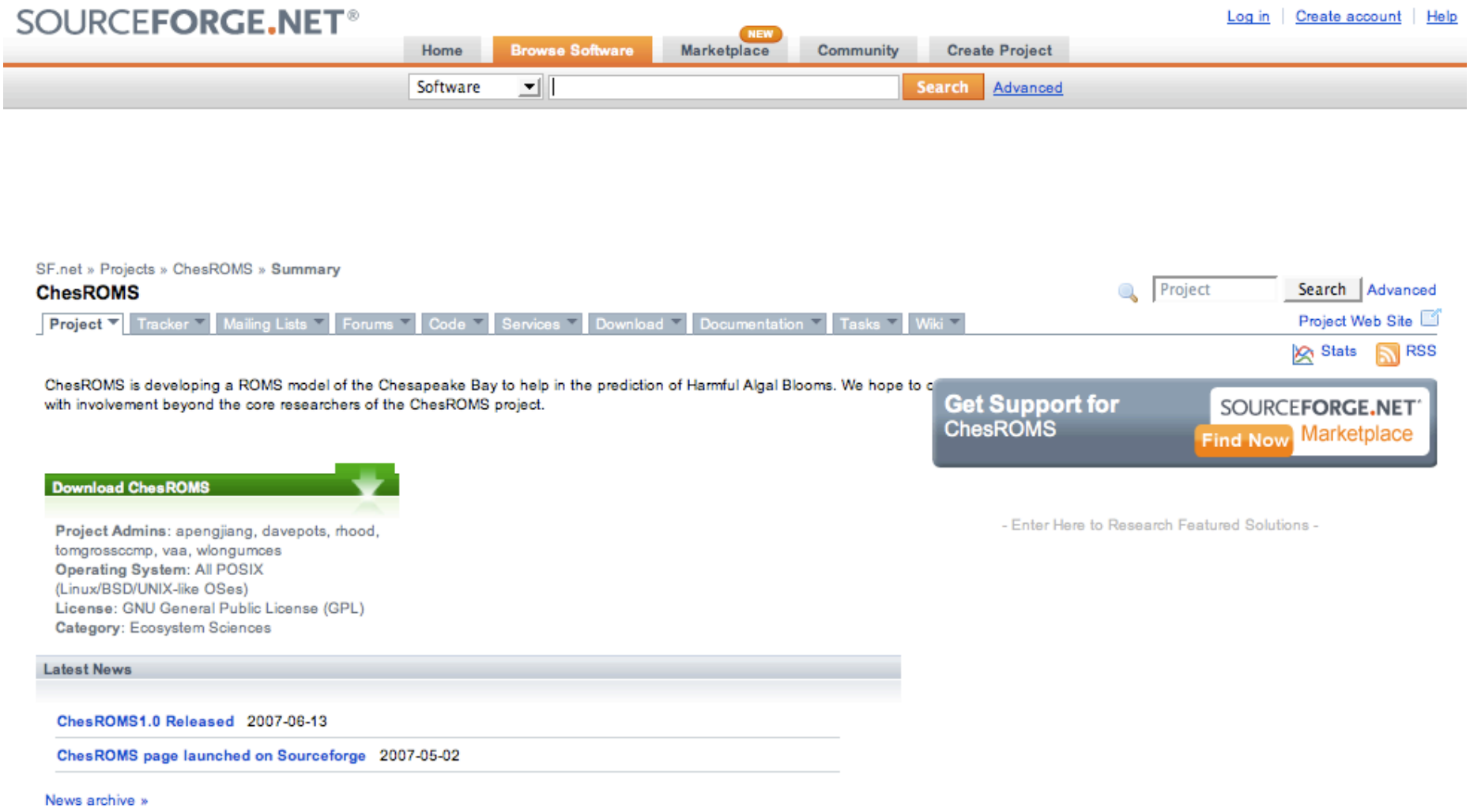
Documentation
- Installation Guide
- How to use SVN

ChesROMS Discussion
- Open Discussion
- Help

ChesROMS Tracker
- Bugs
- Feature Requests
- Patches
- Support Requests

ChesROMS Code
- SVN
- SVN Browse
- SVN Statistics

SourceForge support



Community Modeling Defined:

Community Modeling Defined:

- A community model is an open source component that is constructed and/or improved through the combined efforts of a “community” of individuals working together to help develop, debug, calibrate, document, run and use the model. These individuals often include both developers and users, and may be distributed among a number of different institutions, organizations and geographic locations.

Community Modeling Defined:

- A community model is an open source component that is constructed and/or improved through the combined efforts of a “community” of individuals working together to help develop, debug, calibrate, document, run and use the model. These individuals often include both developers and users, and may be distributed among a number of different institutions, organizations and geographic locations.

Community Modeling Defined:

- A community model is an open source component that is constructed and/or improved through the combined efforts of a “community” of individuals working together to help develop, debug, calibrate, document, run and use the model. These individuals often include both developers and users, and may be distributed among a number of different institutions, organizations and geographic locations.

- Community modeling is a process of supporting, linking and integrating community models, modules and data sets

CCMP umbrella

- HSPF - Chesapeake Bay Program watershed model
- ChesROMS - nowcast and forecast of 3D hydrodynamic circulation, temperature and salinity, sediment transport, biogeochemical and ecosystem states
- C3PO - Chesapeake 3D Physical Oceanographic model
- POMChes - implementation of the Princeton Ocean Model
- SME with LHEM - Spatial Modeling Environment & Library of Hydro-ecological Modules
- CE-QUAL-ICM - Cerco's 4000 cell model
- PIHM

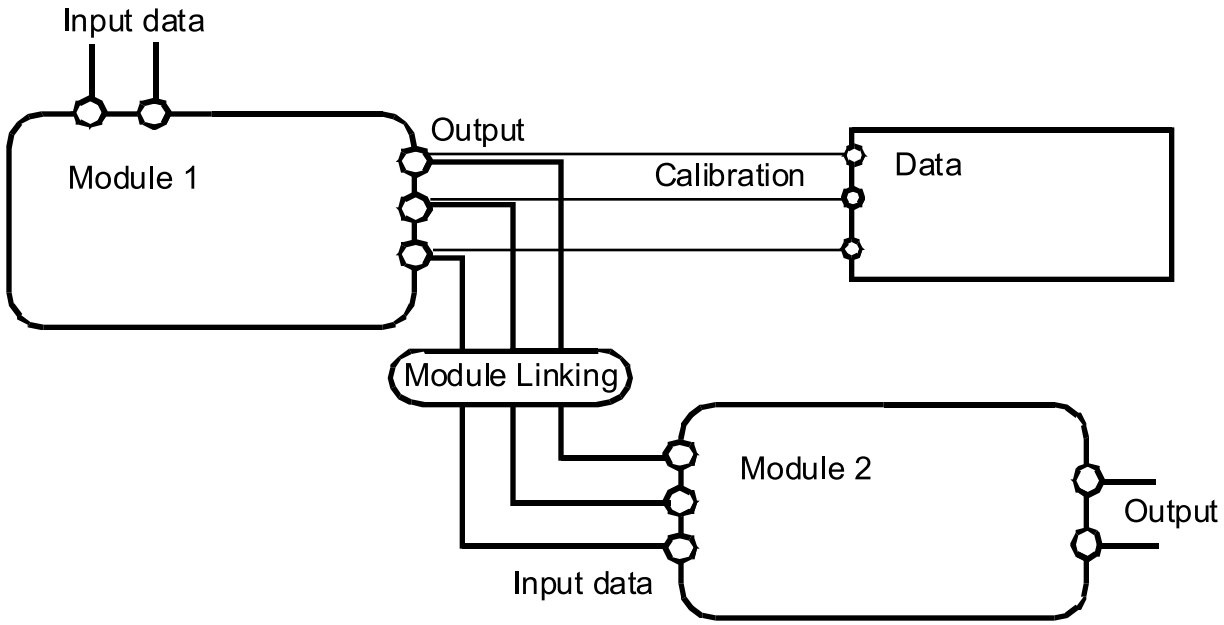
- In progress:
  - SPARROW

Open Source Software vs. Community Modeling

- Smaller numbers of players
- Smaller number of users
- Product is more sophisticated and difficult to use
- Open research is more than open programming
- More sophisticated documentation and steeper learning curve
- Uncertain funding mechanisms
- Ambitions? Credits?
- Lack of infrastructure
- Different modeling paradigms

Linking models

- How to link models?
- How to make them interchangeable and interoperable?



Modularity

Modularity

- **Decomposability criterion - a module should be an independent, stand-alone submodel, that can be analyzed separately**

Modularity

- **Decomposability criterion - a module should be an independent, stand-alone submodel, that can be analyzed separately**
- **Composability criterion - modules can be put together to represent more complex systems**

Modularity

- Decomposability criterion - a module should be an independent, stand-alone submodel, that can be analyzed separately
- Composability criterion - modules can be put together to represent more complex systems
- Decomposability is mostly attained in the conceptual level

Modularity

- Decomposability criterion - a module should be an independent, stand-alone submodel, that can be analyzed separately
- Composability criterion - modules can be put together to represent more complex systems
- Decomposability is mostly attained in the conceptual level
- Composability of modules is mostly a software problem

Modularity

- Decomposability criterion - a module should be an independent, stand-alone submodel, that can be analyzed separately
- Composability criterion - modules can be put together to represent more complex systems
- Decomposability is mostly attained in the conceptual level
- Composability is mostly a software problem

The federation approach - wrappers that enable modules to publish their functions and services using a common high-level interface specification language (CORBA, 1996)

Modularity

- Decomposability criterion
  independent, stand-alone
  separately
- Composability criterion - n
  represent more complex s
- Decomposability is mostly attained in the conceptual level
- Composability of modules is mostly a software problem

The specification approach - embed modules within the context of a specific modeling environment that provides all the software tools essential for simulation development and execution (Maxwell, 1999)

Linking models

- OpenMI (Open Modeling Interface) - a modeling interface standard for hydrologic applications
- Supported by a variety of software tools for migration, linking, performance monitoring and visualization of results
- Delft Hydraulics, DHI (Danish Hydraulics Institute), Wallingford Software
- Centered on the Microsoft Windows platform, including the .NET framework and C#
- It is not intended for high-performance computing
- Much of the core functionality of OpenMI is also being implemented in Java.

Linking modules

- CCA (Common Component Architecture) - DOE-funded "coupling framework"
- Achieves language interoperability using a tool called Babel
- Babel currently supports rapid communication between components written in C, C++, F77, F90, F95, F03, Java and Python.
- Supports parallel computation and many different operating systems but does not support Windows
- CCA should be able to utilize OpenMI components written in Java.

- OMS - Object Modeling System - USDA

Software developer vs. modeler

Software engineer:

- Exponential growth of computer performance offers unlimited resources for the development of new modeling systems.
- Models can be built from blocks connected over the web and distributed over a network of computers

Modeler:

- A model is useful only as a simplification of reality and needs profound understanding of the system to be built.
- Modeling process is of value.
- A model should tell us more about the system, than just the data available

Software developer vs. modeler

Software engineer:

- Exponential growth of computer performance offers unlimited resources for the development of new modeling systems.

New languages and development tools appear faster than their user-communities manage to develop.

Modeler:

- A model is useful only as a simplification of reality and needs profound understanding of the system to be built.
- Modeling process is of value.
- A model should tell us more about the system, than just the data available

Software developer vs. modeler

Software engineer:

- Exponential growth of computer performance offers unlimited resources for the development of new modeling systems.

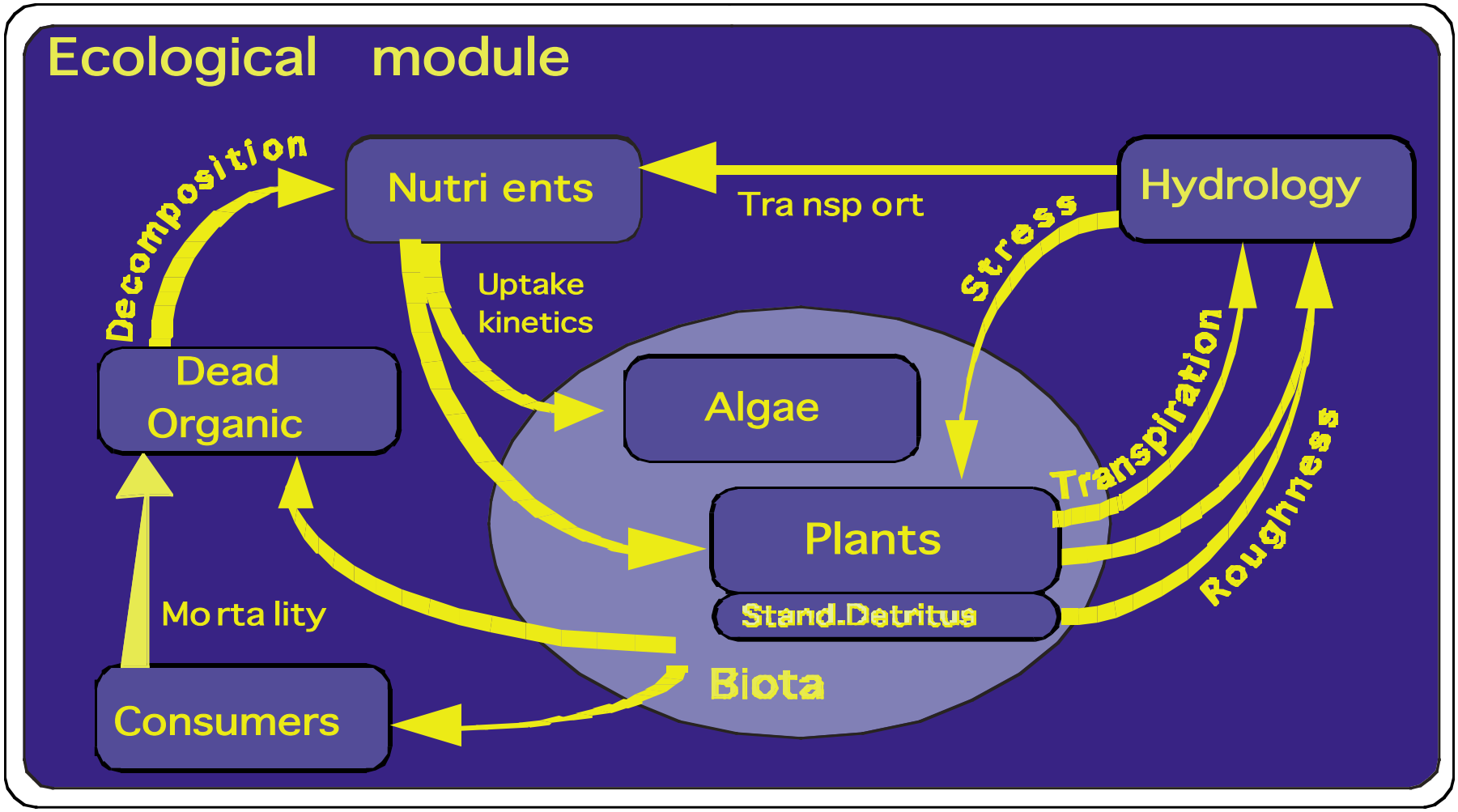
New languages and development tools appear faster than their user-communities manage to develop.

Modeler:

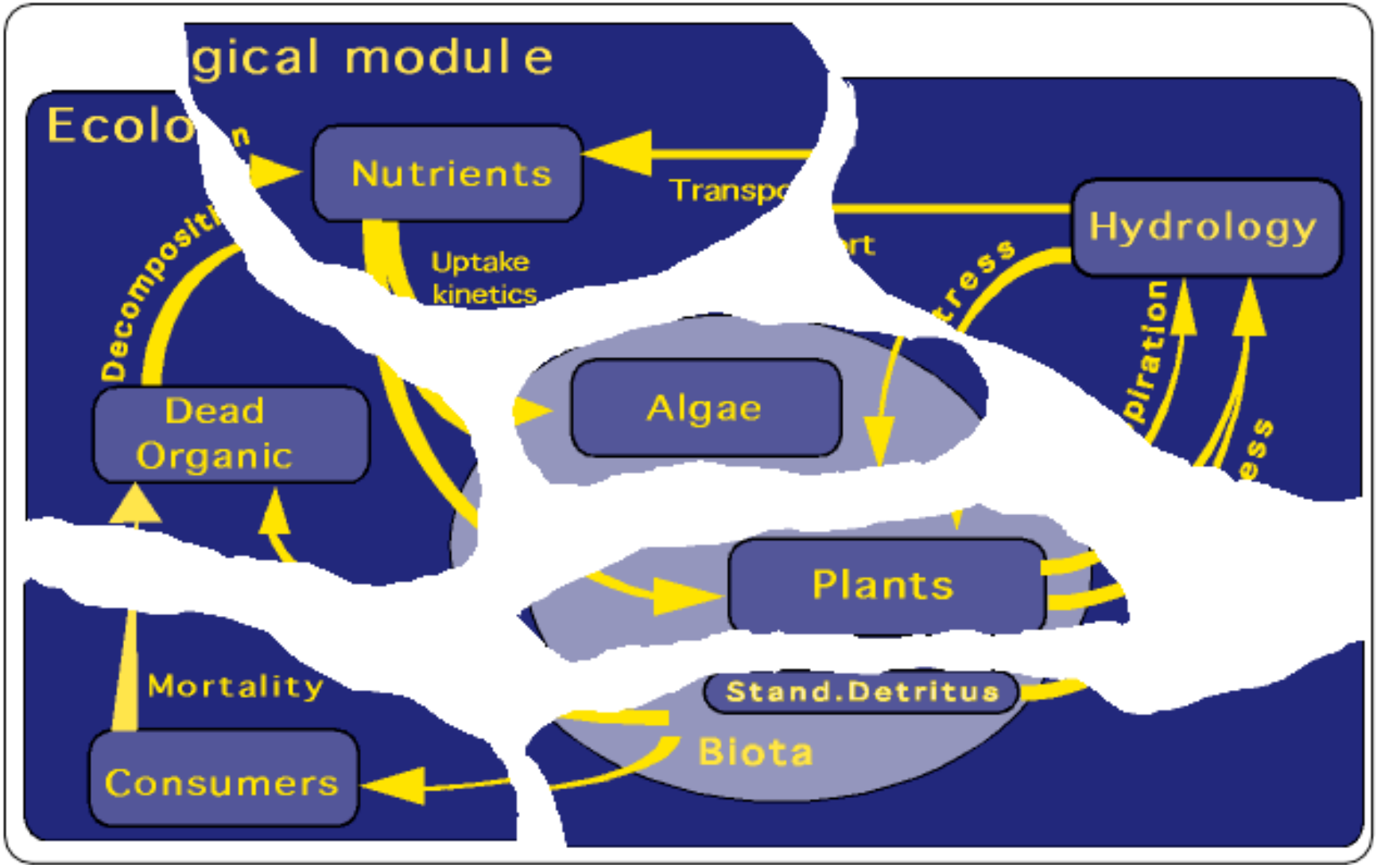
- A model is useful only as a simplification of reality and needs profound understanding of the system to be built.
- Modeling process is of value.
- A model should tell us [illegible] than just the data available

It takes forever to develop and test a model

General Ecological Model (GEM)



Modular approach - Modular Ecological Model



Interoperability needs

Interoperability needs

- Meta data for models

Interoperability needs

- Meta data for models
- Temporal synchronization

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization
- Scale issues

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization
- Scale issues
- Vocabularies and ontologies

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization
- Scale issues
- Vocabularies and ontologies
- Role of data

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization
- Scale issues
- Vocabularies and ontologies
- Role of data
- Calibration, verification, validation

Interoperability needs

- Meta data for models
- Temporal synchronization
- Spatial synchronization
- Structural synchronization
- Scale issues
- Vocabularies and ontologies
- Role of data
- Calibration, verification, validation
- Role of the user

Participatory modeling

Participatory modeling

- Companion modeling, mediated modeling, shared vision planning…

Participatory modeling

- Companion modeling, mediated modeling, shared vision planning…
- Participatory modeling is the process of incorporating stakeholders, often including the public, and decision-makers into the modeling process to support decisions involving complex environmental questions

Participatory modeling

- Companion modeling, mediated modeling, shared vision planning…
- Participatory modeling is the process of incorporating stakeholders, often including the public, and decision-makers into the modeling process to support decisions involving complex environmental questions
- More open and integrated planning processes is a way to avoid potential conflict, misunderstanding and even litigation

Participatory modeling

- Companion modeling, mediated modeling, shared vision planning…
- Participatory modeling is the process of incorporating stakeholders, often including the public, and decision-makers into the modeling process to support decisions involving complex environmental questions
- More open and integrated planning processes is a way to avoid potential conflict, misunderstanding and even litigation
- A platform for integrating scientific knowledge with local knowledge

Participatory modeling

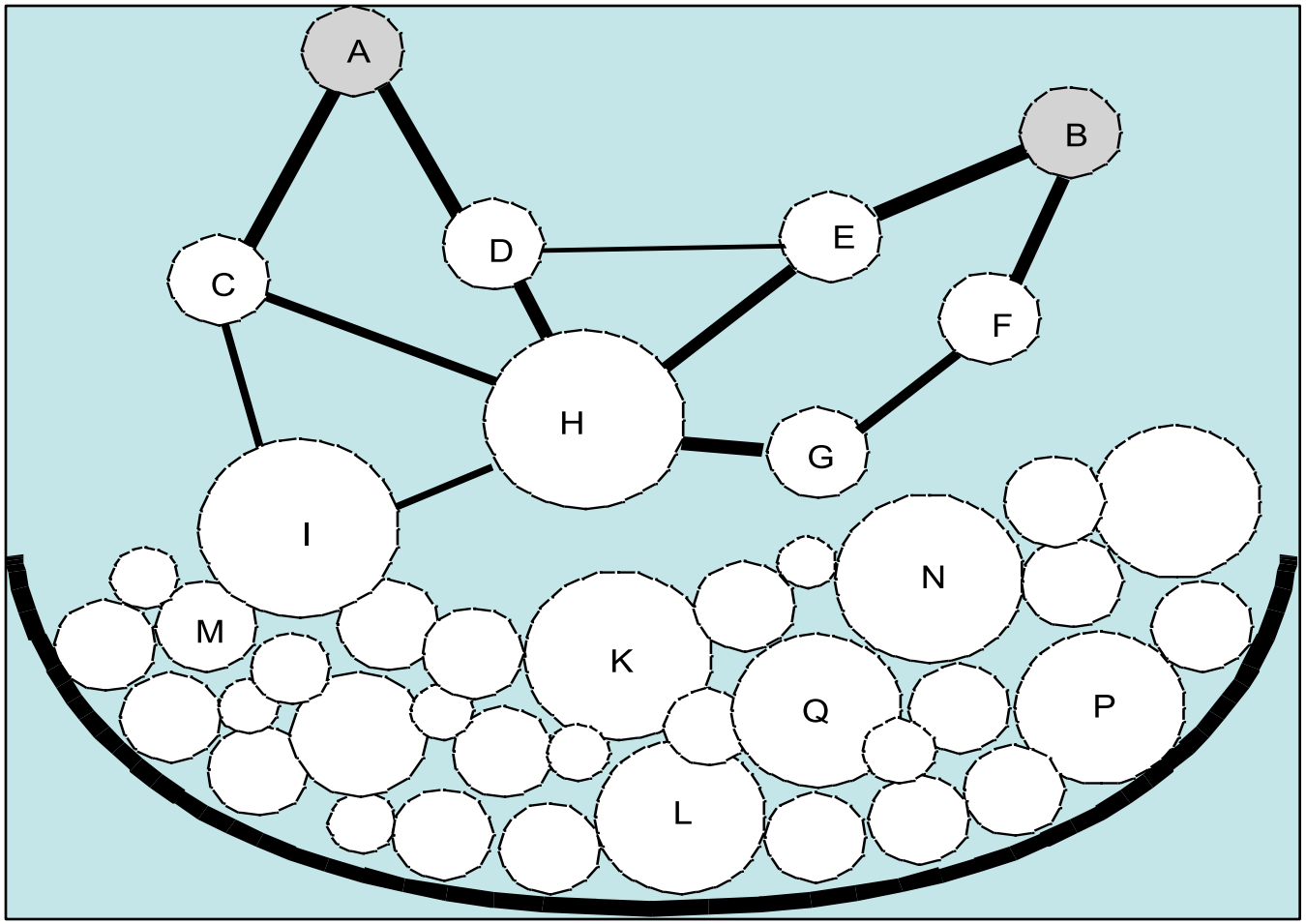
- Companion modeling, mediated modeling, shared vision planning…
- Participatory modeling is the process of incorporating stakeholders, often including the public, and decision-makers into the modeling process to support decisions involving complex environmental questions
- More open and integrated planning processes is a way to avoid potential conflict, misunderstanding and even litigation
- A platform for integrating scientific knowledge with local knowledge
- Goal driven

Participatory modeling

- Companion modeling, mediated modeling, shared vision planning…
- Participatory modeling is the process of incorporating stakeholders, often including the public, and decision-makers into the modeling process to support decisions involving complex environmental questions
- More open and integrated planning processes is a way to avoid potential conflict, misunderstanding and even litigation
- A platform for integrating scientific knowledge with local knowledge
- Goal driven
- Modeling as a process

Model constructor

From the pool of available modules the user picks a few focal modules that respond to the issues at stake (say, A and B). These modules in turn are linked to other modules in the pool that provide input to the focal modules. The complexity of a module is defined by the size of the circle, and the thickness of the link defines its reliability. There may be various minimal configurations that provide functionality to the model thus formulated (say, ADEB is the simplest model, whereas ADHEB is probably the most reliable one in terms of quality of links; ACHGFB may turn out to be the optimal one in terms of module quality).

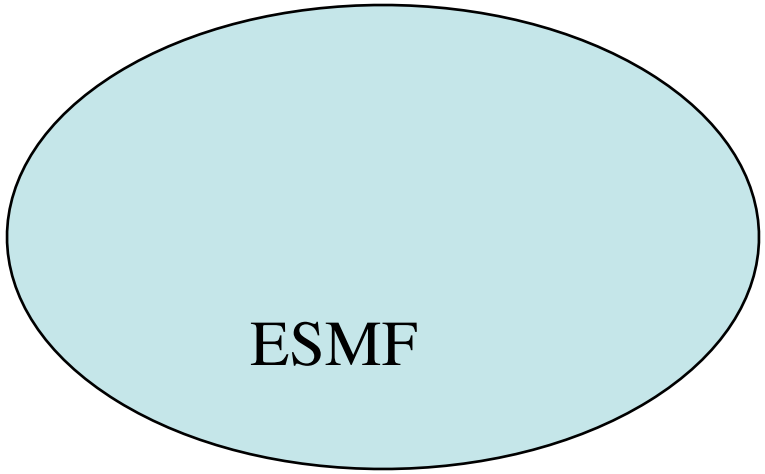


Challenges, concerns

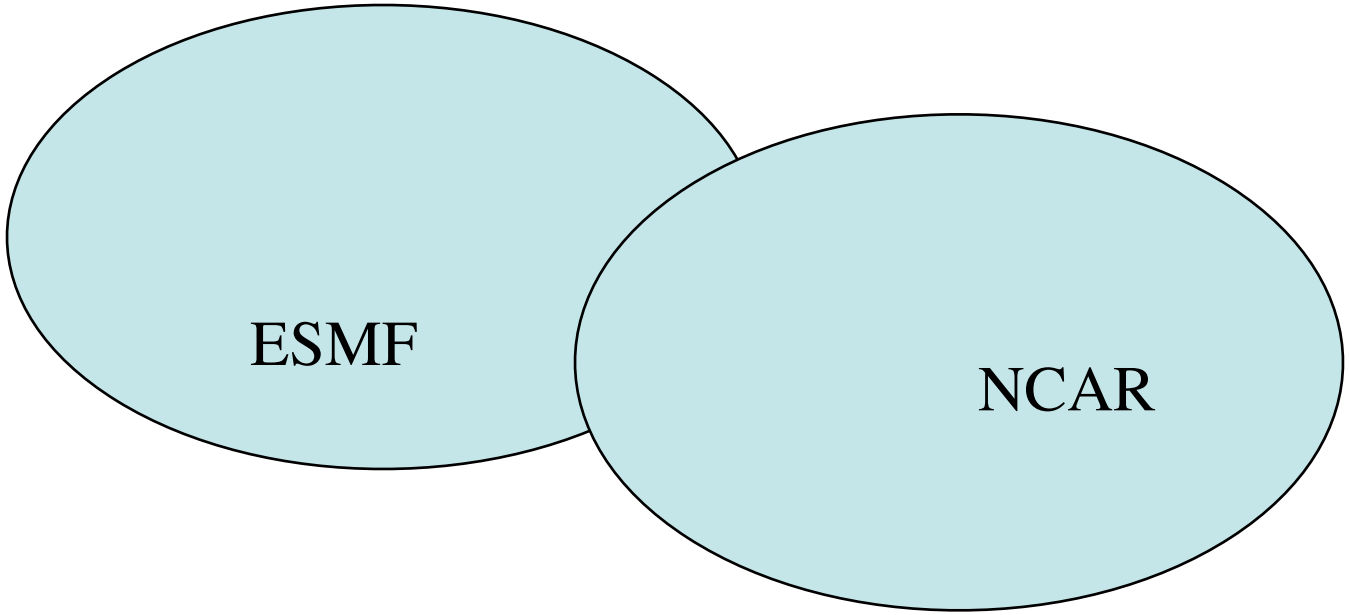
- Community of community models

Communities

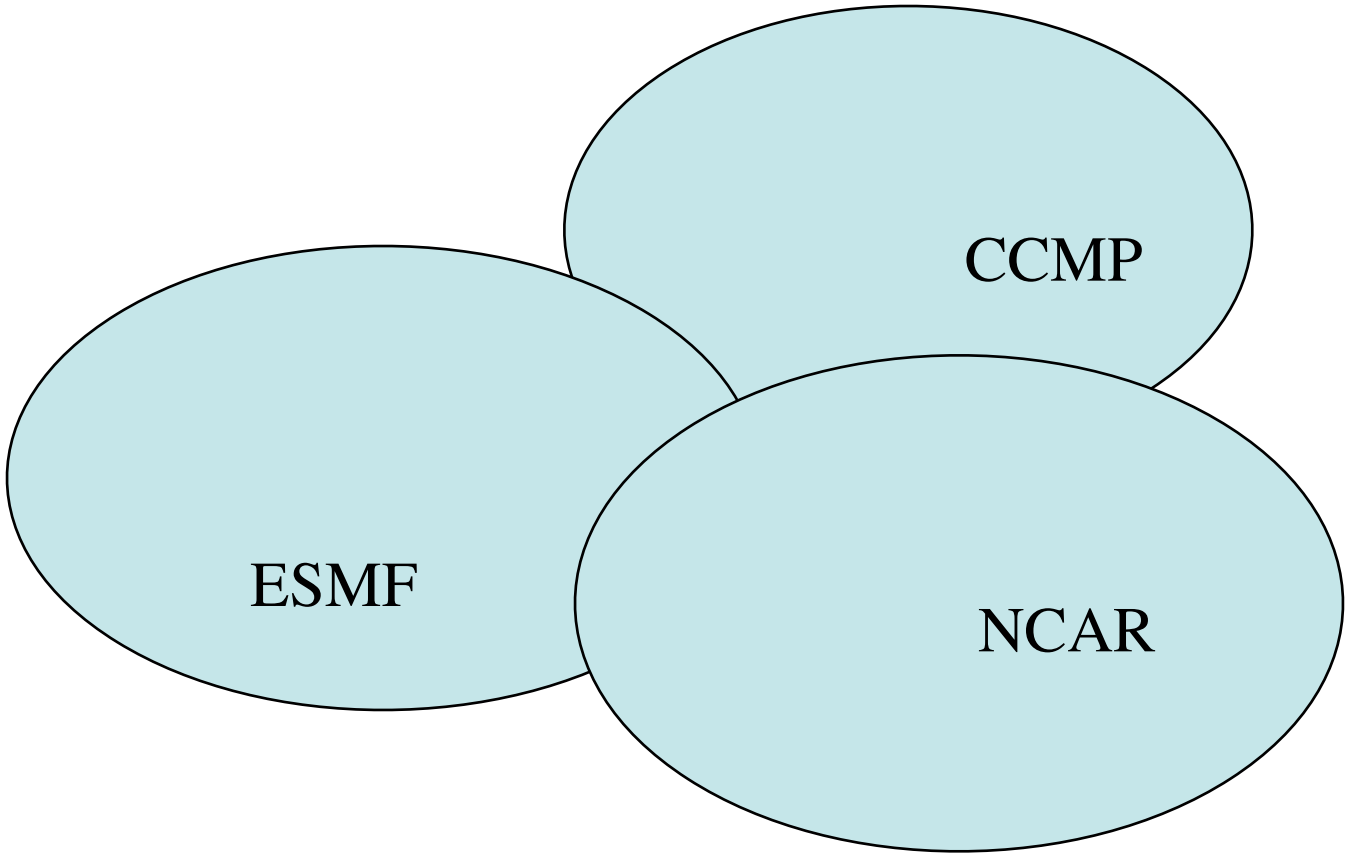
Communities



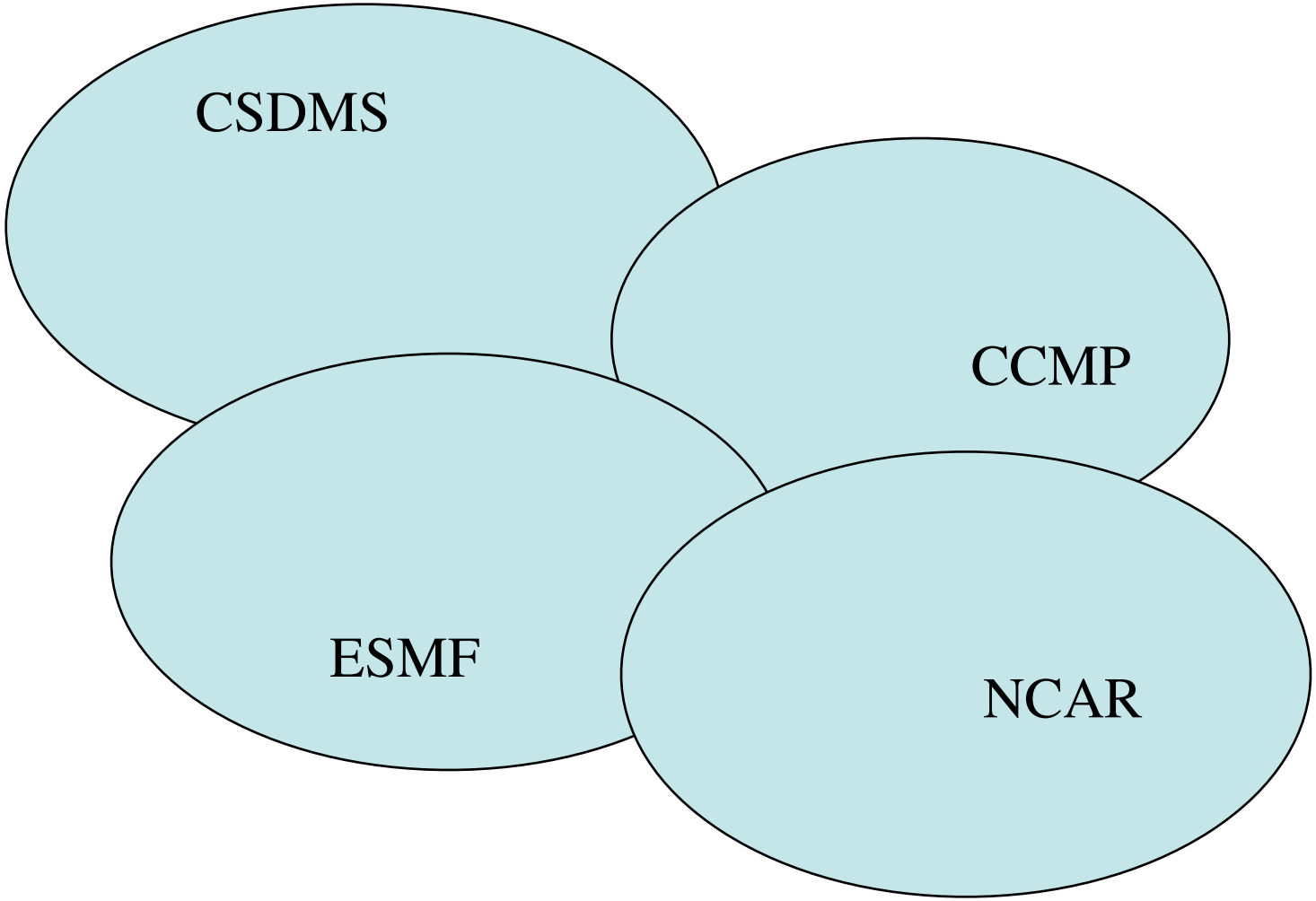
Communities



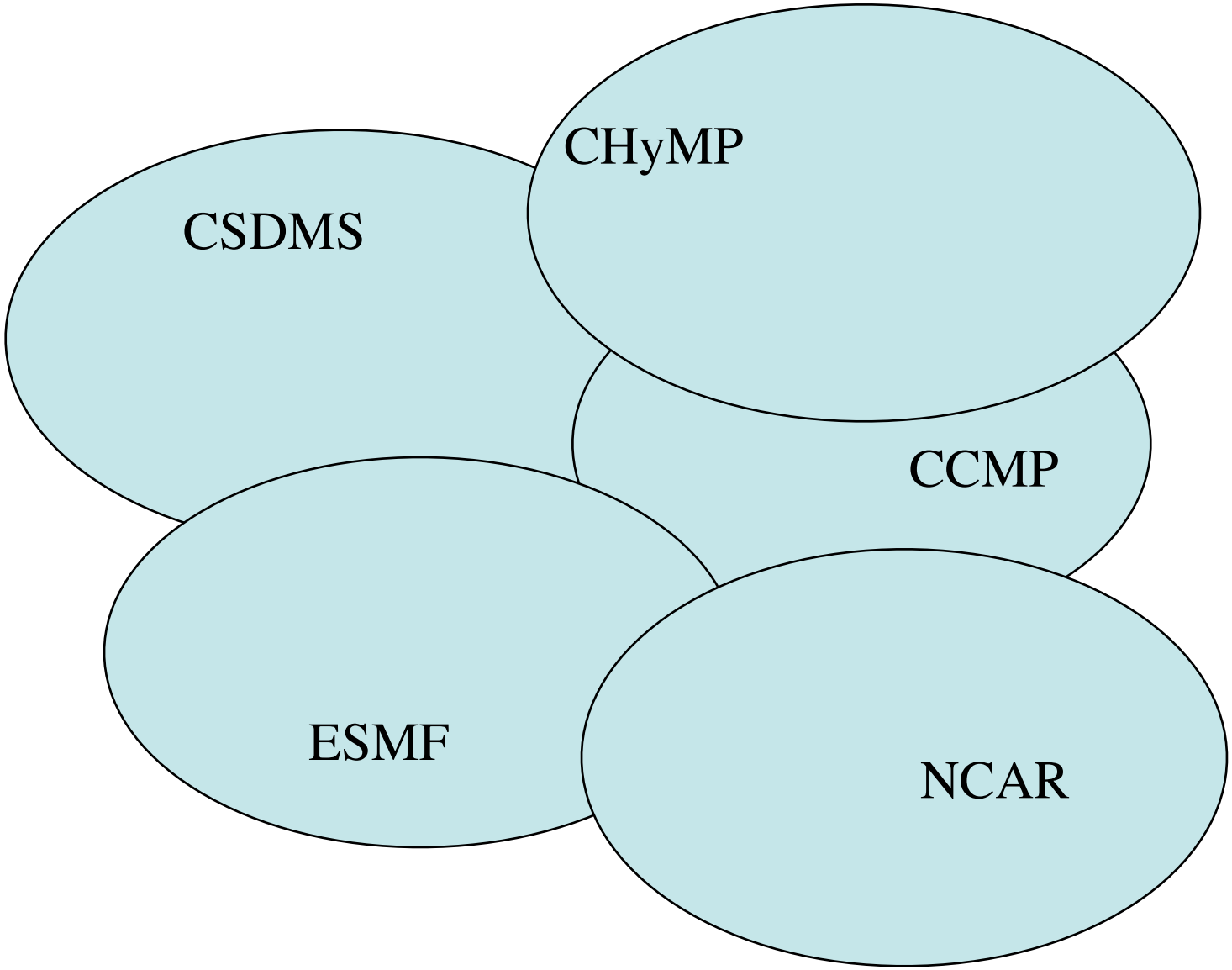
Communities



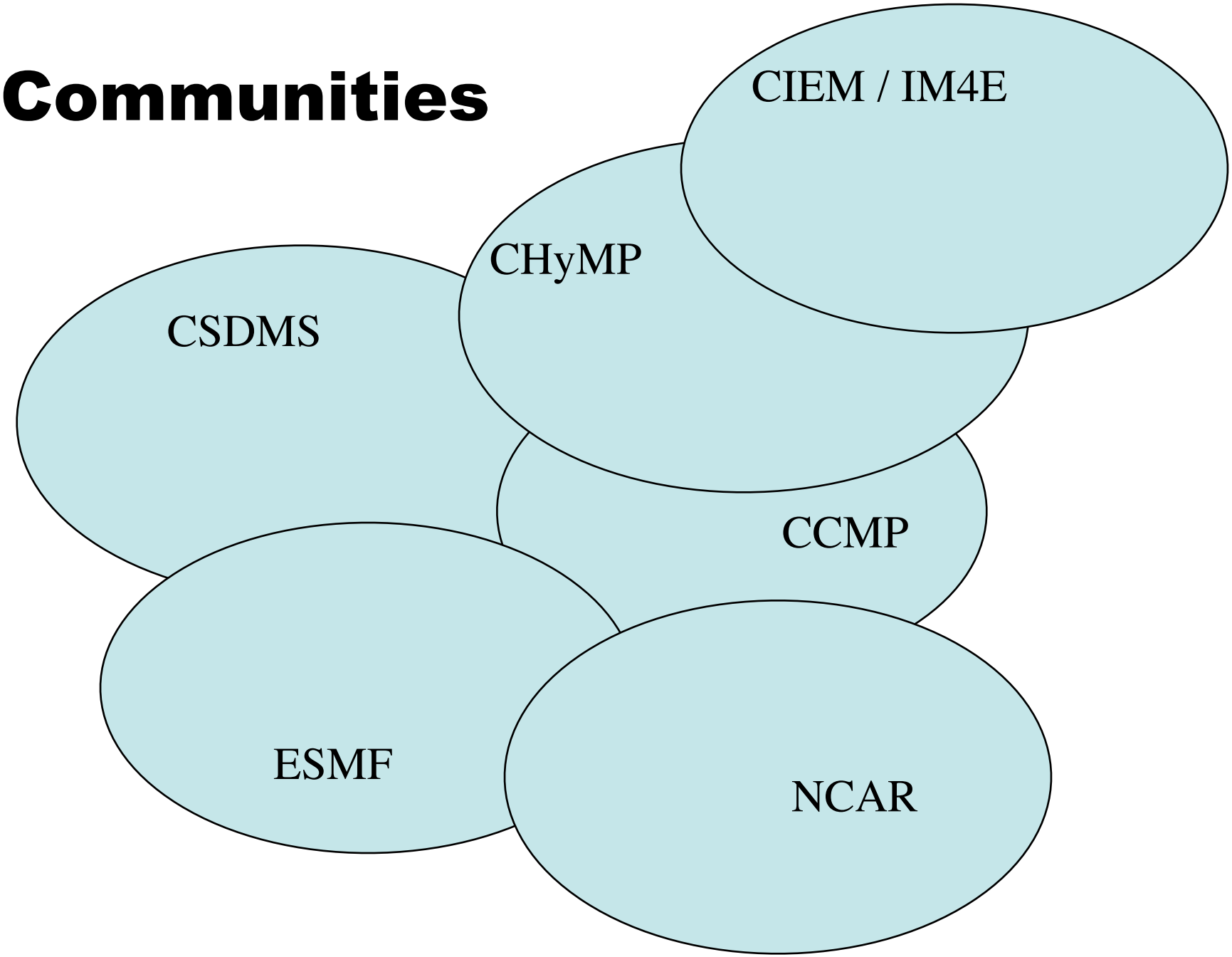
Communities



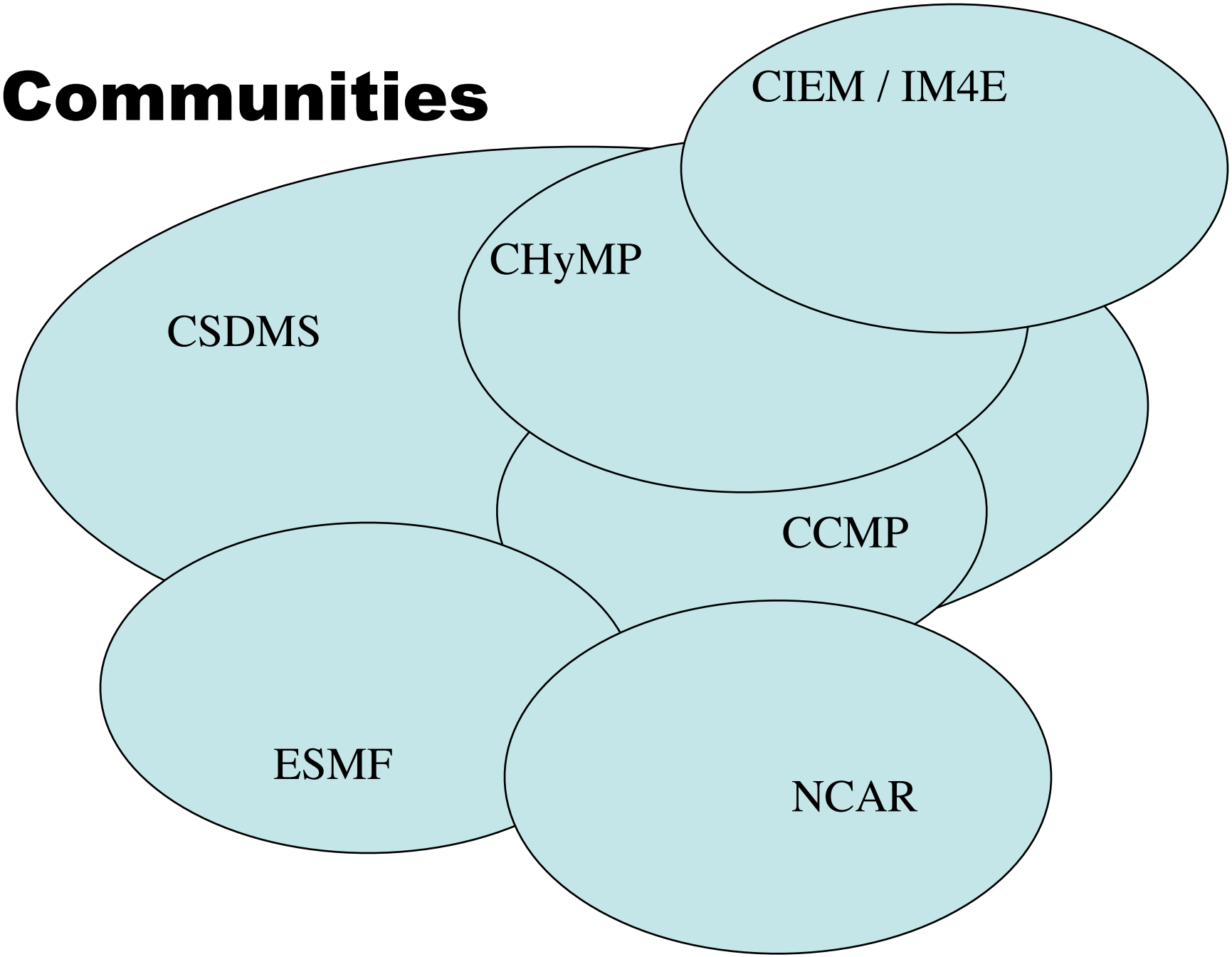
Communities



Communities



Communities



Challenges, concerns

- Branching
- No obligations, but…
  - How do we lure in participants?
  - How is it beneficial for them?
- New modeling paradigms, specific culture and ethics
- Standards, standards, standards

Advantages

- Facilitates integration of effort among multiple institutions, research groups and/or individuals;
- Improves continuity and project survival in face of uncertain funding and institutional support;
- Cuts redundancy. There is less need for "reinventing the wheel";
- Allows scientists to work with software engineers, bridging the cultural and, often, institutional gap between these disciplines;
- Provides the essential link to the user community, offering transparency that promotes user participation and input at early stages of the project and during the testing phase. More users provide better testing, more robust models and more acceptance of the results.

Challenges

- Reward structure is skewed toward publications and away from technical contributions;
- Funding is discontinuous, and not reliably available for support and technical infrastructure;
- Intellectual property policies of universities and private companies are not compatible. Software is often viewed as a competitive advantage for funding and academic honors;
- A “not made here” culture;
- Overhead of “soft” organizations that have no formal hierarchy. The “bazaar” approach does not work well with deadlines and deliverables. Lack of realistic project assessment and clear strategies to deal with conflicts and inefficiencies;
- Difficulties of work across distances and time zones with a diverse group of people;
- Need of transdisciplinary communication among scientists, engineers, users, and decision-makers. All are important segments of the modeling community, but each has its own culture, vocabulary, and objectives.

Technical recommendations for developers and the rest of us

- Adopt existing standards for data, model input and output, and interfaces.
- Develop standards for model conceptualization, formalization and scaling.
- Seek to use/adapt existing tools first before developing your own.
- Provide good documentation, including examples and test cases.
- Establish and use good software development practices that favor transparency, portability, and reusability, and include procedures for version control, bug tracking, regression testing, and release maintenance

International Environmental Modeling and Software Society (IEMSS)

- 300+ members
- Journal “Environmental Modeling and Software” by Elsevier
- Bi-annual conferences (Lugano - Switzerland, Osnabrück - Germany, Burlington - USA, Barcelona - Spain)
- 2010 - Ottawa - Canada