A Brief Introduction to the CSDMS Initiative

Dr. Scott Peckham
Chief Software Architect for CSDMS
October 26, 2007

csdms.colorado.edu

CSDMS
Community Surface Dynamics Modeling System

CUAHSI Fall 2007
Regional Meeting
Boise, Idaho

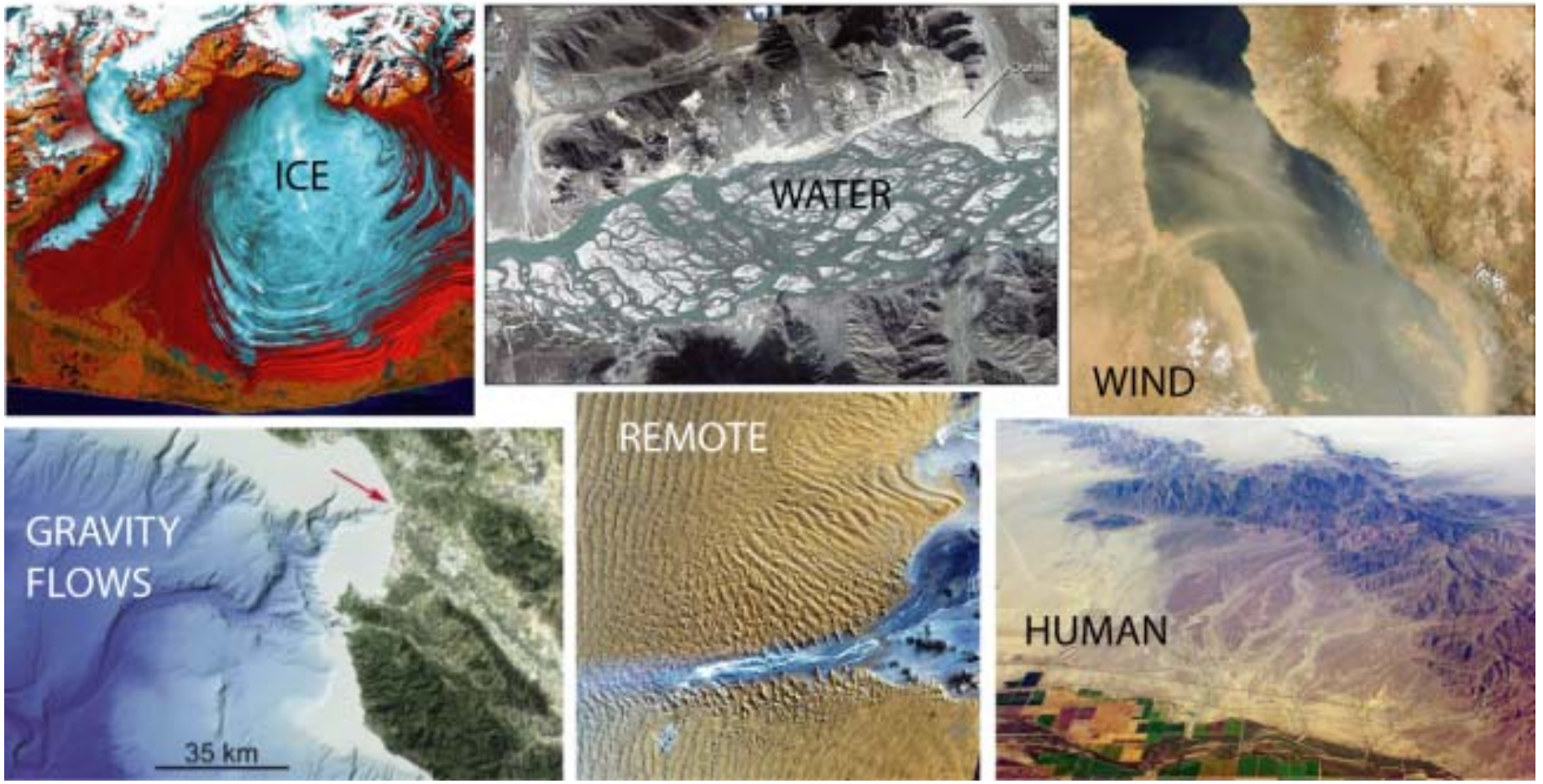
What is CSDMS ?

- Stands for "***Community Surface Dynamics Modeling System***" (acronym can be pronounced "systems" or spelled out)
- A recently-awarded NSF cooperative aggreement, initially for 5 years, with Prof. James Syvitski as PI and Executive Director
- An effort to bring together a diverse community of *surface dynamics modelers* and *model users,* as has been done successfully in other communities (e.g. climate modeling)
- About 140 members (so far) divided into 5 working groups, with governance by Executive & Steering committees, and support from NSF/EAR/OCE, USGS, NOAA, and others
- A facility located at the University of Colorado in Boulder, with a staff of 7 to 10 people

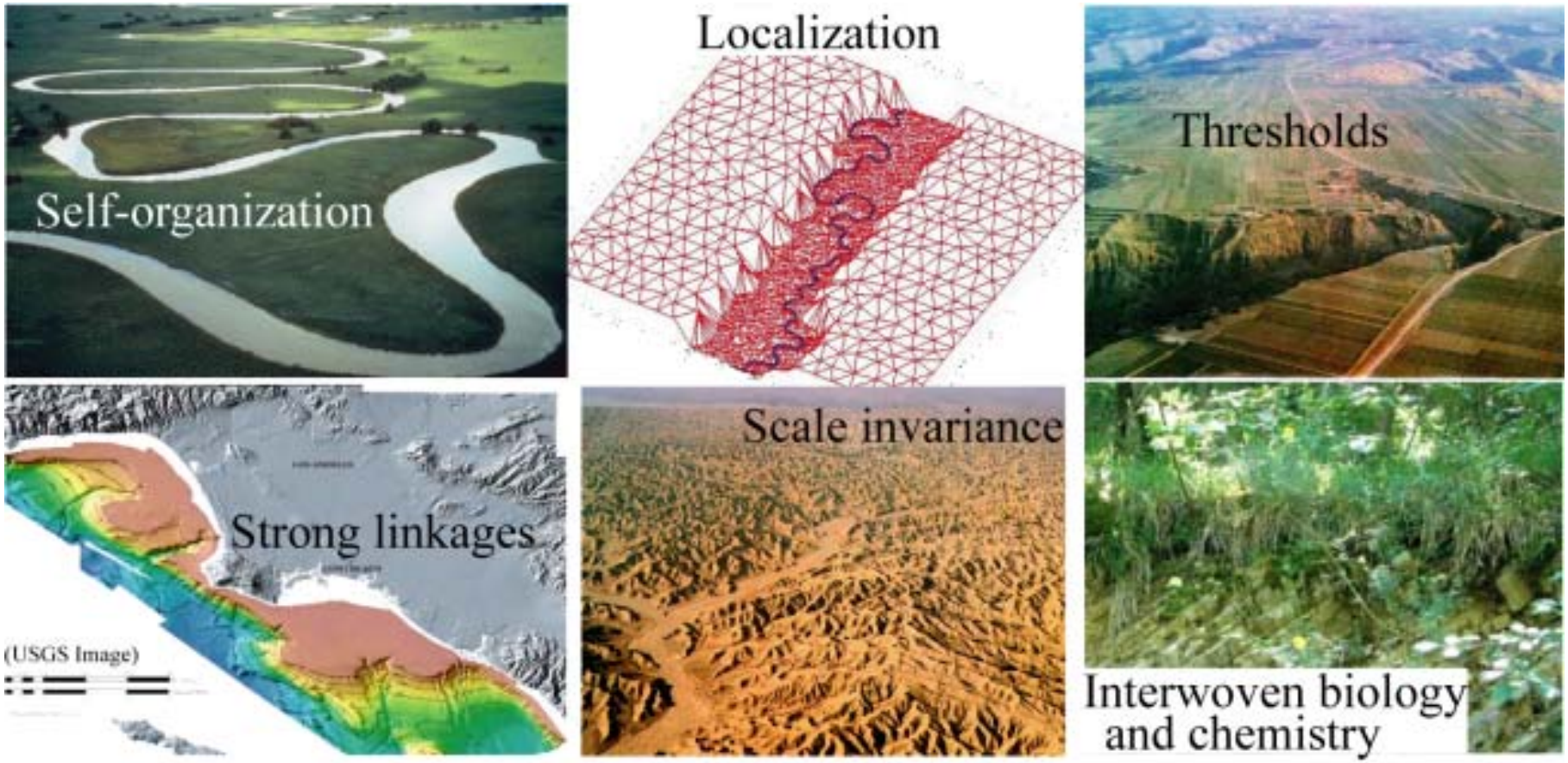
Key Goals of the CSDMS

- Promote code sharing and re-use (open source)
- Develop a review process for contributions
- Promote recognition of contributors
- Develop a “library” of low-level software tools and higher-level models that can be linked as easily as possible into new applications
- Provide resources to simplify the efforts of surface dynamics modelers
- Provide an organized, searchable repository of surface dynamics models and tools

The CSDMS Domain

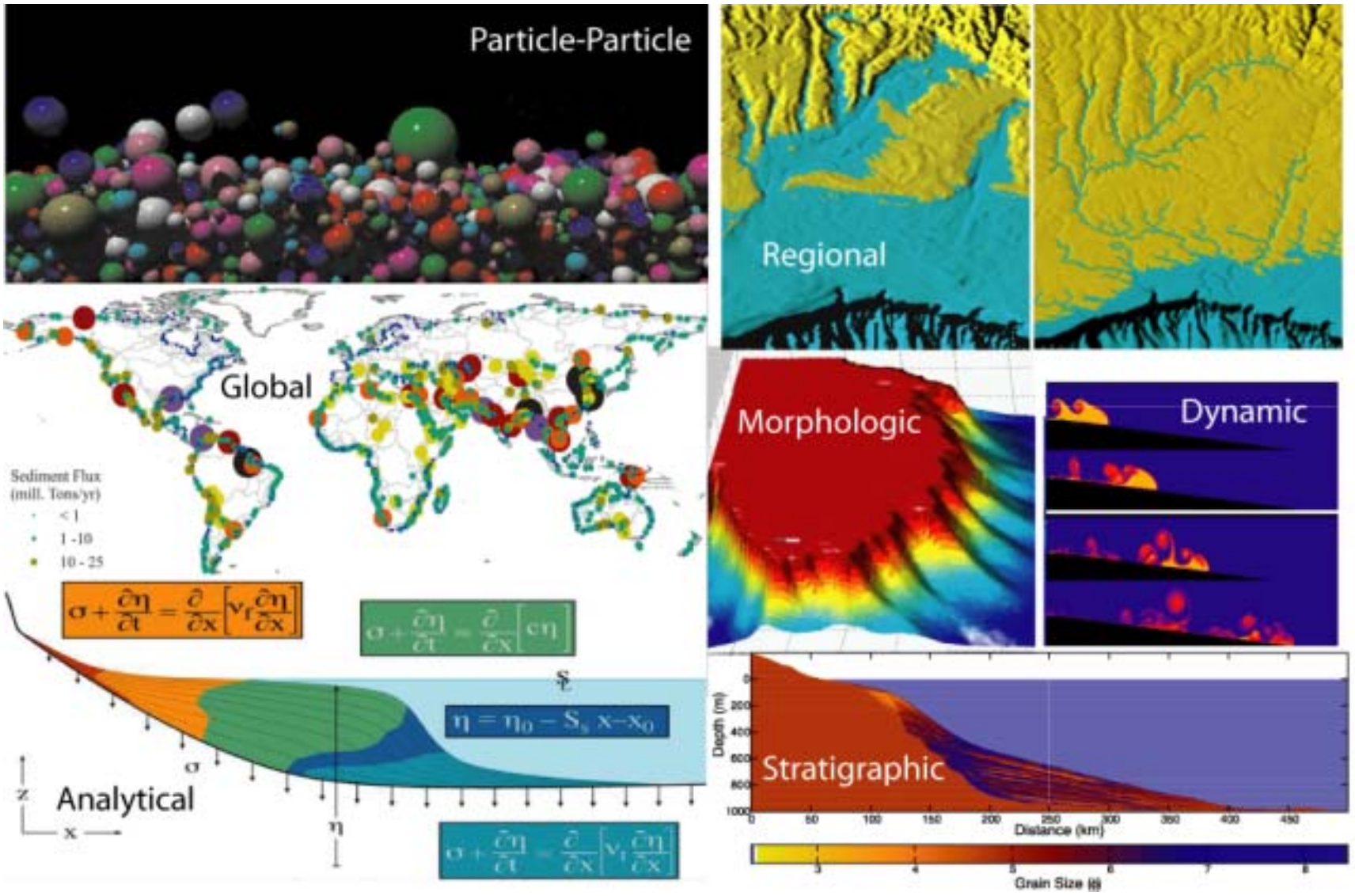


Some of the Key Properties of Surface Systems



Key properties of surface systems

Diversity of Existing Modeling Efforts



Tentative Functional Specs for the CSDMS

Support for multiple operating systems
(especially Linux, Mac OS X and Windows)

Support for parallel (multi-proc.) computation (via MPI standard)

Language interoperability (e.g. CCA is language neutral) to support code contributions written in C, Fortran as well as more modern object-oriented languages (e.g. Java, C++, Python)

Support for both legacy (non-protocol) code and more structured code submissions (procedural and object-oriented)

Should be able to interoperate with other coupling frameworks

Support for both structured and unstructured grids

Platform-independent GUI (e.g. via wxPython)

Large collection of open-source tools

“Coupling Frameworks”

ESMF (Earth System Modeling Framework)

www.esmf.ucar.edu, maplcode.org/maplwiki

PRISM (Program for Integrated Earth System Modeling)

www.prism.enes.org (uses OASIS4)

OpenMI (Open Modeling Interface)

www.openmi.org

CCA (Common Component Architecture)

www.cca-forum.org,

www.llnl.gov/CASC/components/babel.html

Others: GoldSim (**www.goldsim.com**) commercial

FMS (**www.gfdl.noaa.gov/~fms**) GFDL

Overview of ESMF

Widely used by U.S. climate modelers

Based on **Fortran90** (efforts underway for C coupling)

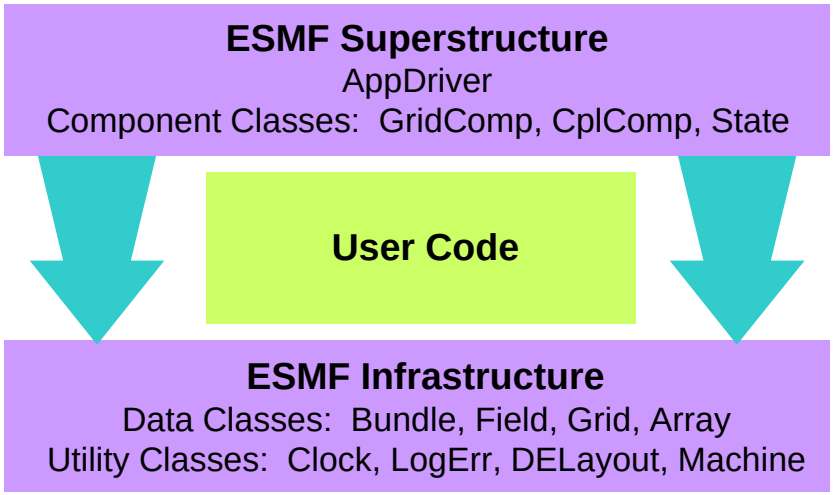
Components follow the **Initialize, Run, Finalize** scheme

Has a new development tool called **MAPL**

Started with NASA, now has buy-in from NOAA, DoD, DOE, NSF. May be adopted by CCSM; see:

www.ccsm.ucar.edu/cseg/Projects/Working_Groups/soft/esmf

- Parallel-computing friendly (MPI)
- Compatible with PRISM & CCA.
- Many useful tools in its Infra-structure & Superstructure
- Mainly structured grids so far



Overview of OpenMI

Developed by **hydrologic community** in Europe with corporate buy-in (e.g. Delft Hydraulics)

Based on **Microsoft's C#** (similar to Java) and support for Java is under development by HydroliGIS (Italy)

Components follow the **Initialize, Run, Finalize** scheme

Emphasizes support for data formats (e.g. WML)

Currently incompatible with non-Windows computers, so language and platform specific

Designed for a single-processor environment

Funding future is currently uncertain

Does not seem to have the maturity or buy-in of ESMF & CCA.

Overview of CCA

Widely used at DOE labs (e.g. LLNL, ANL, Sandia) for a wide variety of projects (e.g. fusion, combustion)

Language neutral; Components can be written in C, C++, Fortran 77-95, Java, or Python; supported via a compiler called **Babel**, using SIDL / XML metadata

Interoperable with ESMF, PRISM, MCT, etc.

Has a development tool called **BOCCA (& Eclipse)**

Similar to CORBA & COM, but science app support

Can be used for single or multiple-processor systems, distributed or parallel, MPI, high-performance (HPC)

Structured, unstructured & adaptive grids

Has DOE / SciDAC (www.scidac.gov) funding support

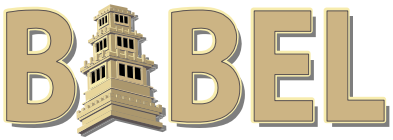
Overview of CCA (continued)

CBSE = Component-Based Software Engineering

Support for **Java & Python** makes it possible to add components with GUIs, graphics or network access anywhere in the application (e.g. via **wxPython**). Python code can be compiled to Java with **Jython**. (See **www.jython.org** for details)

Python is used by Google and is the new ESRI scripting language. It is entirely open-source and a large number of components are available (e.g. XML parser) Currently has over one million users and growing.

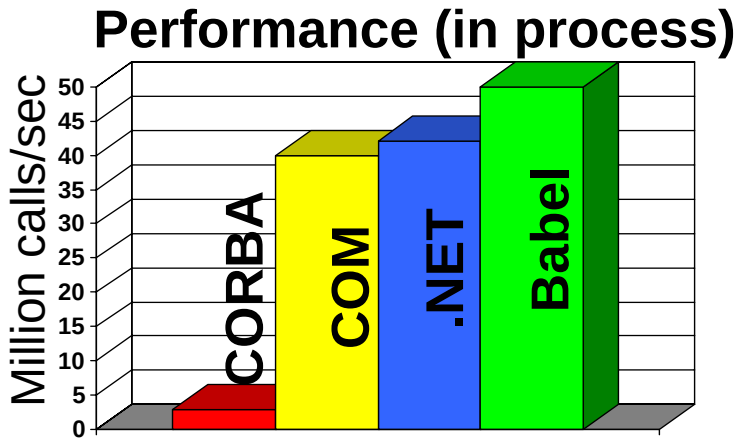
CCA: The Babel Tool



is Middleware for HPC

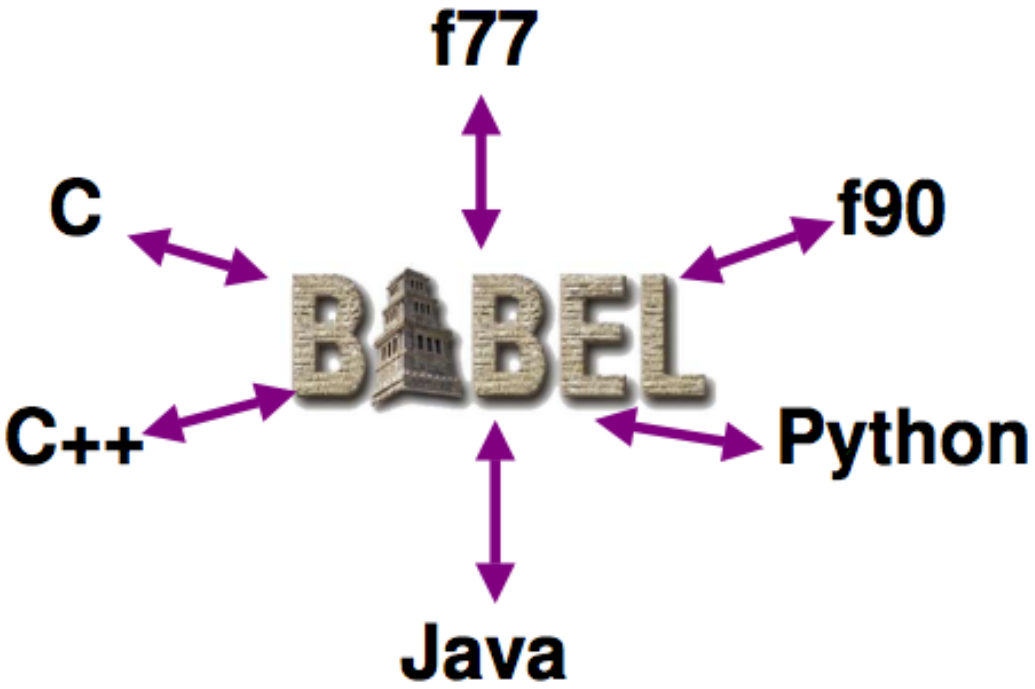
2006

"The world's most rapid communication among many programming languages in a single application."



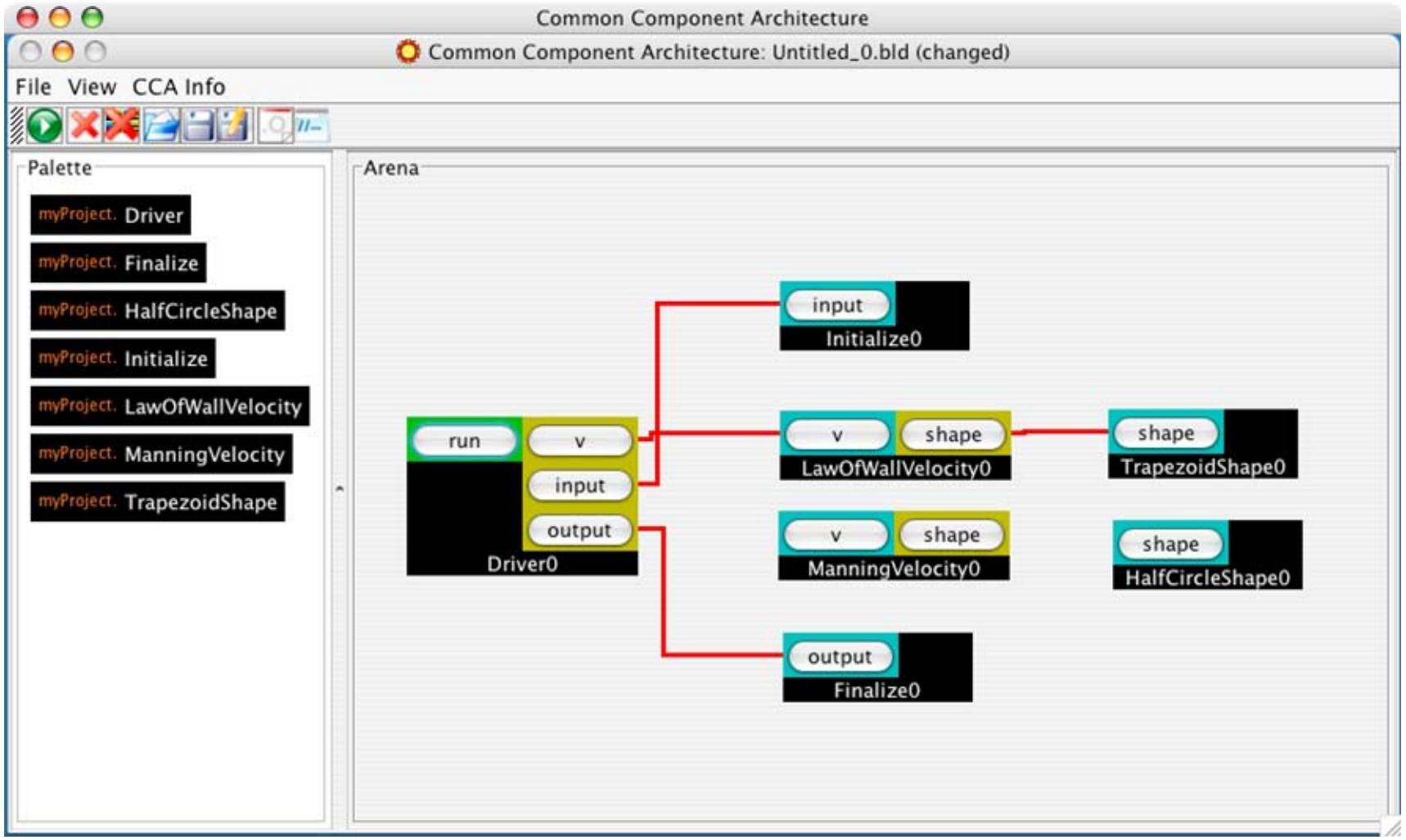
	CORBA	COM	.NET	Babel
BlueGene, Cray, Linux, AIX, & OS X	No	No	No	Yes*
Fortran	No	Limited	Limited	Yes
Multi-Dim Arrays	No	No	No	Yes
Complex Numbers	No	No	No	Yes
Licensing	Vendor Specific	Closed Source	Closed Source	Open Source

CCA: The Babel Tool



Language interoperability is a powerful feature of the CCA framework. Components written in different languages can be rapidly linked with hardly any performance cost. This allows us to "shop" for open-source solutions (e.g. libraries), gives us access to both procedural and object-oriented strategies (legacy and modern code), and allows us to add graphics & GUIs at will.

CCA: The Ccaffeine Tool



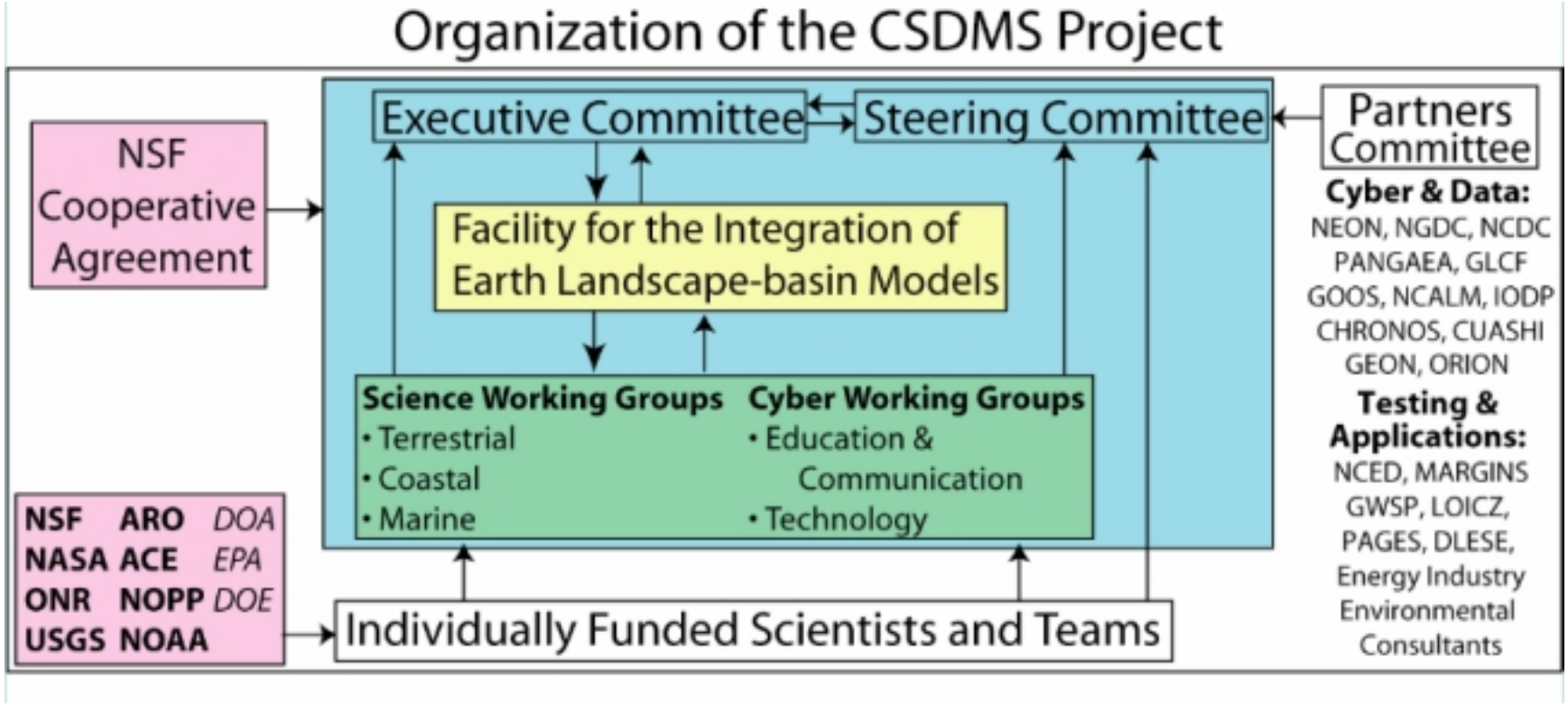
The CCA framework includes a graphical tool called **Ccaffeine** for linking together model components.

Conclusions

Component-based software design is taking off in several different scientific communities.

There are a large number of relatively mature and increasingly user-friendly tools and frameworks for developing and linking component-based software.

The CCA framework looks particularly promising and is likely to be adopted by the CSDMS. However, it is interoperable with ESMF and would also be interoperable with a Java version of OpenMI.
The Babel tool could potentially add support for C#.


Organization of the CSDMS Project
NSF Cooperative Agreement
Executive Committee
Steering Committee
Facility for the Integration of Earth Landscape-basin Models
Science Working Groups
• Terrestrial
• Coastal
• Marine
Cyber Working Groups
• Education & Communication
• Technology
NSF ARO DOA
NASA ACE EPA
ONR NOPP DOE
USGS NOAA
Individually Funded Scientists and Teams
Partners Committee
Cyber & Data:
NEON, NGDC, NCDC
PANGAEA, GLCF
GOOS, NCALM, IODP
CHRONOS, CUASHI
GEON, ORION
Testing & Applications:
NCED, MARGINS
GWSP, LOICZ,
PAGES, DLESE,
Energy Industry
Environmental
Consultants