



---

# Space Weather Modeling Framework Component Interface Workshop

David Chesney and Ovsei Volberg  
Software Engineering  
Center for Space Environment Modeling  
The University of Michigan



# Objective

---

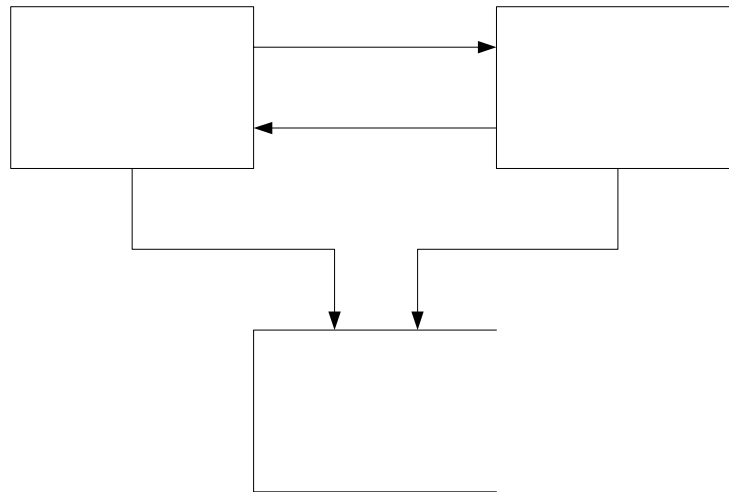
The SWMF provides a flexible Sun-Earth simulation framework serving the Sun-Earth community.

In its fully developed form, the SWMF will comprise a series of interoperating models of physics domains, ranging from the surface of the Sun to the upper atmosphere of the Earth.



# Approach

## Historic: Tightly Coupled Code

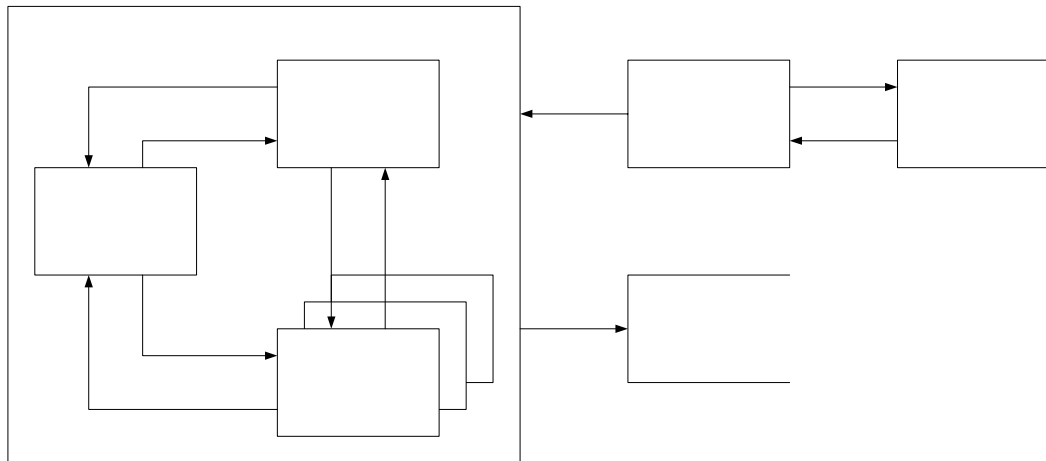




# Approach

Present: Framework w/ Small Number of Components

- Prototype Phase
- Milestone 7I, Aug03

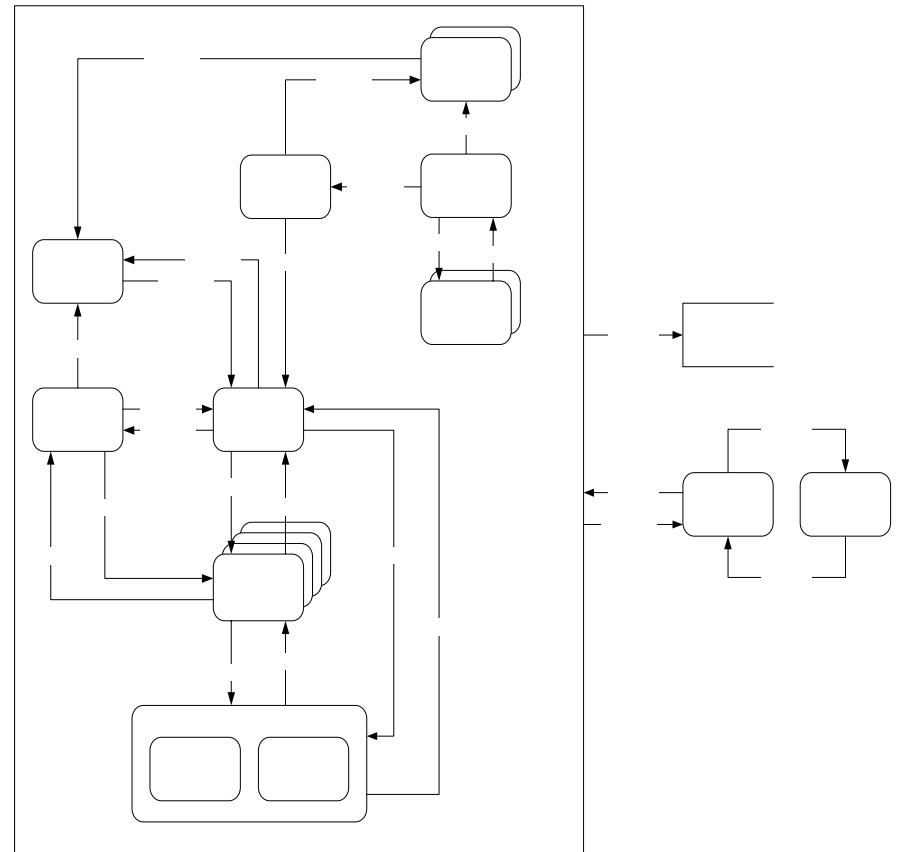




# Approach

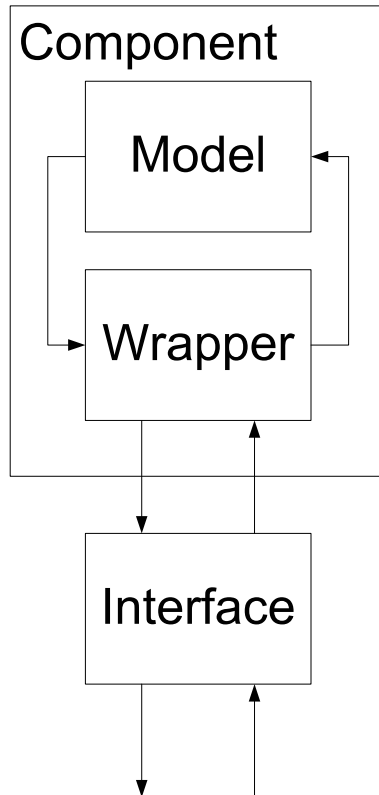
## Future: Full Framework

- Final Phase
- Milestone 10J, Jul04





# Physics Component (Language)

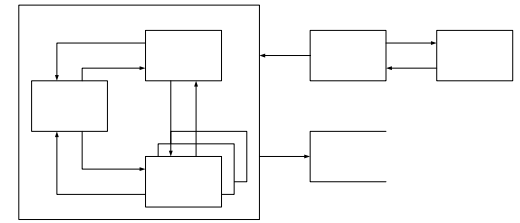


Physics **Component** is:

- Physics **Model**
  - ★ Solution code for physics domain
  - ★ Language: F77, F90
- Physics Model **Wrapper**
  - ★ Unit conversion
  - ★ Data transformation
  - ★ Language: same as Physics Model
- Physics Model **Interface**
  - ★ Data exchange between physics modules
  - ★ Completes mappings between two coupled physics modules
  - ★ Receives and processes signals from CON
  - ★ Monitors the execution of coupled physics modules
  - ★ Language: 7I: F90; 10J: F90, C++



# SWMF (Language)



SWMF	Description	Language: Milestone 7I	Language: Milestone 10J
Physics Model Interface	See previous slide	F90	C++/F90
Control	<ul style="list-style-type: none"> <li>• Distribution of PMs to PEs</li> <li>• Time-stepping of individual PMs</li> <li>• Coupling between PMs</li> </ul>	F90	C++/F90
GUI	<ul style="list-style-type: none"> <li>• Choice of PMs</li> <li>• Parameter submission</li> <li>• Monitor execution</li> <li>• View results</li> </ul>	HTML, XML, Javascript, PERL	Same
Output File (OUF)	<ul style="list-style-type: none"> <li>• Output file format</li> </ul>	Proprietary	Proprietary, HDF-5



# Language Independence

Language neutrality? **Yes** and No

- **Yes:**

- ★ Follow OOP paradigms

- ★ Consistent definition of functions/methods:

- 🌐 constructor

- 🌐 destructor

- 🌐 get

- 🌐 set

- 🌐 solve

- 🌐 stop



# Language Independence

---

Language neutrality? Yes and **No**

- **No:**

- ★ Phase 7I written in F90



# System Requirements (Execution)

Machine	Location	OS	Fortran Compiler	C/C++ Compiler	Usage
Beowulf (16 PEs)	CSEM – UMICH	RedHat Linux v7.2	NAG v4.0a	gcc/g++	execution
Beowulf (120 PEs)	CSEM – UMICH	RedHat Linux v7.2	NAG v4.0a	gcc/g++	execution
SGI Origin (16 PEs)	CAC – UMICH	IRIX 6.5	MIPSpro 7	MIPSpro 7	execution
SGI Origin (1024 PEs)	NASA – Ames	IRIX64	MIPSpro	MIPSpro	execution
Compaq	NASA – GSFC	Tru64 Unix	f90	cxx	execution



# System Requirements (Development and GUI)

Machine	Location	OS	Fortran Compiler	C/C++ Compiler	Usage
AMD Athalon (2 PEs)	CSEM – UMICH	RedHat Linux v7.2	NAG v4.0a	gcc/g++	develop- ment
AMD Athalon (1 PE)	CSEM – UMICH	RedHat Linux v7.2			GUI
Varies	Varies				remote user



# Execution Model

- Milestone 7I: *Sequential – Parallel* model of execution
  - ★ Physics modules executing on multiple processors (i.e., parallel), however, only one physics module is executing at any given time (i.e., sequential)
- Milestone 10J: *Sequential – Parallel* model of execution acceptable, however, *Concurrent – Parallel* model encouraged
  - ★ Physics modules executing on multiple processors (i.e., parallel), with multiple physics modules executing at same time (i.e., concurrent)



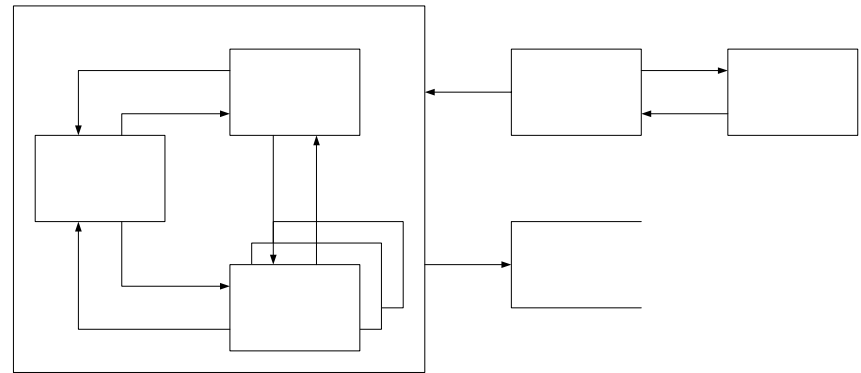
# Component/Component Relationship

In general:

- Component/Component
  - ★ peers
  - ★ cannot spawn each other
  - ★ no side effects

However:

- CON/Component
  - ★ client/server





# Persistence

---

- Component (Physics Model) Persistence
  - ★ fixed at compile time
- Compute Resource (Processing Element) Persistence
  - ★ fixed at compile time



# Standards

(The Wild Frontier between Responsibility and Authority)

- Interoperability Document
  - ★ Initialized in parallel configuration.
  - ★ Activated by Control Module for data exchange with other component.
  - ★ Queried for run status, output, and input data.
  - ★ Perform time synch and conversion, and performance profiling.
  - ★ Provide error trapping and report to the Control Module.
  - ★ Write state to a restart file.
  - ★ Read input data from files (path to files is externally controlled/set).
  - ★ Write output data to files (path to files is externally controlled/set).
  - ★ To be registered by the Control Module.
- Data Naming Standard



# Target User Community

---

- Modelers in space physics community and space weather forecasters



# Target Component Authors

---

## Modelers in the space physics community

### Phase 7I

- Inner magnetosphere model: Rice University

### Phase 10J

- Energetic particle model: University of Arizona
- Eruptive event generator: University of New Hampshire
- Radiation belt model: Rice University
- Kinematic heliosphere model: UCSD



# Summary

---

- Implementation of framework based upon phased development
- Milestone 7I (Prototype Phase) due in August, 2003
  - ★ decouple existing code
  - ★ recouple with standard interfaces
  - ★ centralize control
  - ★ show ability to substitute alternative physics components
- Milestone 10J (Final Phase) due in July, 2004
  - ★ add many additional physics components