INTEGRATED FRAMEWORKS FOR EARTH AND SPACE WEATHER SIMULATION

Timothy Killeen and Cecelia DeLuca*
National Center for Atmospheric Research, Boulder, Colorado

Tamas Gombosi, Gabor Toth, and Quentin Stout
University of Michigan, Ann Arbor, Michigan

Charles Goodrich
Boston University, Boston, Massachusetts

Alan Sussman
University of Maryland, College Park, Maryland

Michael Hesse
Goddard Space Flight Center, Greenbelt, Maryland

ABSTRACT

Simulations of Earth and space weather require the representation and coupling of distinct physical domains in a flexible, computationally efficient manner. There is increasing call to interface Earth and space models, as the interplay of phenomena between these domains is an active topic of basic research, and a key factor in operational prediction systems. Software frameworks have been developed in both the Earth and space communities to promote code reuse, coupling and interface standards, and high performance. These include the Earth System Modeling Framework (ESMF)[1], developed for use in numerical weather prediction, climate prediction, and other Earth science applications; the Space Weather Modeling Framework (SWMF)[2], developed for high performance space weather modeling and self-consistently coupling models from the low solar corona to the Earth ionosphere; and the tools assembled by the Center for Integrated Space Weather Modeling (CISM)[3], which focus on modeling systems in which flexibility and ease of adoption are critical factors. In this paper we present a technical strategy and pilot projects that combine these three frameworks into an integrated system, in a way that takes advantage of the strengths of each.

1 INTRODUCTION

In this paper we will describe three Earth and space software frameworks, and examine how interfaces between them may be developed for the benefit of the research community. The three frameworks are a software system created by the Center for Integrated Space Weather Modeling, a NSF Science and Technology Center, the Space Weather Modeling Framework (SWMF), from the Center for Space Environment Modeling at the University of Michigan, and the Earth System Modeling Framework (ESMF), a multi-agency effort based at the National Center for Atmospheric Research (NCAR). ESMF focuses on climate and weather, while the two space weather frameworks focus on simulations that extend from Earth's upper atmosphere to the solar corona.

The structure of the paper is as follows. Section 2 describes frameworks and related software concepts, the reasons for developing frameworks, and the motivation for integrating frameworks in the Earth and space weather domains. Sections 3, 4, and 5 describe the ESMF, SWMF, and CISM frameworks, and highlight some of their similarities and differences. Section 6 summarizes and compares the capabilities of the three frameworks. Section 7 describes a set of pilot projects designed to explore integration options. Conclusions are in Section 8.

Descriptions of the frameworks and physics domains are drawn directly from existing documents prepared by the respective project teams [1,2,3,4,5].

2 BACKGROUND AND MOTIVATION

Frameworks have emerged in the Earth and space domains as a response to the increasing complexity of models and computing systems. Comprehensive Earth and space models encompass multiple, interacting physics domains. The software representing these domains is often developed by different groups, and must be coupled together to form composite applications. The applications are computationally intensive and execution time directly affects the ability of the models to deliver useful results. Challenges posed by computing platforms include the transience and variety of architectures and compilers, issues of distributed and shared memory, and a lack of adequate general purpose middleware for routine modeling functions. Frameworks are intended to promote ease of model development, integration, extension, modification, and use. There is

---

* Corresponding author address: Cecelia DeLuca, NCAR, P.O. Box 3000, Boulder, Colorado, 80307-3000; email: cdeluca@ucar.edu.

[1] http://www.esmf.ucar.edu
[2] http://www.csem.engin.umich.edu/SWMF
[3] http://www.bu.edu/cism/

also a cultural motivation, as the process of framework development and increased code interoperability within a science domain can help to create a more cohesive and collaborative community.

2.1 Definitions

An object-oriented framework is formally defined as an application that is architected but incomplete, and can be customized for specific scenarios [e.g., 6]. The idea is to allow application developers to focus on content rather than design strategies and computing details. However, Earth and space modelers face limitations in their ability to develop object-oriented software due to the omnipresence of legacy Fortran codes, and face architectural challenges due to applications that use the same elements in a variety of contexts (for example, an atmospheric model used for data assimilation research, ensemble NWP forecasting, and a seasonal prediction application). Thus for the Earth and space domains, a software framework may be defined *in practice* as a set of tools used for the development, assembly, configuration, and execution of an application.

For complex applications, frameworks are increasingly structured using component-based design principles. A component is a unit of software composition that has a coherent function, and a standard calling interface and behavior. Components can be assembled to create multiple applications, and different implementations of a component may be available [e.g., 7]. Following Toth et al [2], we refer to a specific instance of a particular component as a *component version*. For Earth and space frameworks, components may be physics domains, or functions such as couplers or I/O systems. A framework may be comprised of a specified set of components, or it may be a methodology for assembling applications that allows for different types of components and different drivers. The SWMF and the European climate framework PRISM (Programme for Integrated Earth System Modeling) [8] fall into the former category, while ESMF and CCA (the Common Component Architecture) [9], which is a general framework for high performance computing, fall into the latter.

Component-based frameworks are used to implement *models* and *modeling systems*. We associate a model with a particular numerical representation of a physical domain, a specified set of one or more component versions that together describe the domain, and a driver. We assume that a model is configurable and executable. A modeling system is not tied to a particular implementation of physics, but to a discipline or problem (i.e., integrated forecasts, climate variability). A modeling system may have one or more drivers and a large set of component versions, with multiple swapping options for many components. Models may be identified within modeling systems, either explicitly by name or through specification of a particular set of component versions that is physically meaningful.

2.2 Framework Interactions

It is plausible to implement the component versions included in the CISM model in the SWMF framework and vice versa, since both represent comparable physical processes and extents. However, there is a development cost associated with adapting a component version for a particular framework, that varies depending on the framework and the component version. Other issues of compatibility arise between CISM and SWMF frameworks because of differences in how components and inter-component interactions are defined.

It is important to recognize that the grid representations and coupling services required for a model may not be implemented in a particular framework. This is the reason why space weather models cannot be readily run using the ESMF framework. The climate and weather domain relies largely on simple, logically rectangular grids, and transformations between components tend to be from one 2D surface to another (i.e., the atmosphere/ocean surface). Space weather grids and inter-component transformations are far more complex, and ESMF does not currently support many of the required structures and operations. As a general rule, moving a component version between frameworks often requires work on the both the component version and the framework, and may require changes to other component versions as well.

However, where components are defined similarly, or the framework scope is very general, it is possible to structure component versions so that they can run in multiple frameworks, accruing benefits from each. We cite two examples.

In climate, the major US framework is ESMF; in Europe, it is PRISM. PRISM follows a multi-executable approach much like the CISM effort, while ESMF follows a structured, single executable approach much like SWMF. Working together, ESMF and PRISM have developed a strategy whereby codes can be set up to run in either framework, and have demonstrated this with the MOM4 ocean model at the NOAA Geophysical Fluid Dynamics Laboratory. The advantage to instrumenting MOM4 in this way is that it makes the code available to a very broad set of modelers, while retaining a single source for code maintainability.

A second example involves ESMF and CCA. These teams have demonstrated that it is possible to wrap an ESMF component, without modification, as a CCA component and run it using a CCA driver [10]. Here the CCA framework layer provides additional services that ESMF does not, such as a configuration GUI, the potential to run applications over the Grid, and the ability to swap in components at runtime.

2.3 Motivation for the Integration of Earth and Space Frameworks

The motivation for interfacing ESMF-based Earth simulations to space simulations and vice versa is the natural desire to better express boundary conditions for each domain, and to improve representation of the boundary region. Overall goals are to deepen understanding of physical processes and to produce better forecasts. The ability to exchange component versions between the CISM and SWMF frameworks would afford space weather researchers more opportunities for experimentation and collaboration. There are also a number of upper atmosphere researchers who are interested in implementing their component versions in an ESMF-compatible way.

For the reasons outlined in the previous section, we see that it is currently technically impractical to use a single framework for both the Earth and space domains. (It is also culturally impractical, but we leave those arguments aside.) We also see that there are precedents for combining multiple frameworks. In the following sections, we examine and compare the design and implementation of the ESMF, SWMF and CISM frameworks to better understand what the costs and benefits are of integration.

3 THE SPACE WEATHER MODELING FRAMEWORK (SWMF)

The Space Weather Modeling Framework is used to model the region from the Sun's surface to the upper atmosphere of a planet, usually Earth. The SWMF framework driver is customized for a set of space weather components, and the framework is distributed with a set of component versions that are working and tested [11]. The SWMF modeling system includes multiple options for several components. Configuration options are also offered that enable the user to run physically meaningful subsets of components. There is a SWMF model associated with a full set of active components. Many of the component versions within the SWMF are also models that may be executed standalone.

3.1 Physics Domains

The SWMF integrates numerical models of the Solar Corona, Eruptive Event Generator, Inner Heliosphere, Solar Energetic Particles, Global Magnetosphere, Inner Magnetosphere, Radiation Belt, Ionosphere Electrodynamics and Upper Atmosphere. In the current version of the SWMF, the Eruptive Event Generator and Solar Corona domains are represented by one component.

The Solar Corona, Inner Heliosphere and Global Magnetosphere component versions are based on the University of Michigan's BATS-R-US code [12]. The Solar Corona code uses magnetogram measurements to specify realistic boundary conditions for the potential magnetic field at the Sun [13]. There are two versions of the Eruptive Event Generator, both developed at the University of Michigan. One is based on the Gibson and Lowe flux rope implicit scheme, which assumes that the flux rope exists prior to eruption [14]. In the second version, the flux rope is created during eruption [15]. There are also two versions of the Solar Energetic Particles component. The first is Kota's SEP model, which is based on an implicit scheme and was developed at the University of Arizona [16]. The second version is the Field Line Advection Model for Particle Acceleration (FLAMPA), which uses an explicit scheme, from the University of Michigan [17]. Both the Inner Magnetosphere and the Radiation Belt model are from Rice University. The Radiation Belt component version is the Rice Radiation Belt Model (RRBM), which is not yet fully functional, and the Inner Magnetosphere is the Rice Convection Model (RCM) [18,19]. The Ionosphere Eectrodynamics model is from the University of Michigan [20,21]. There are two versions of the Upper Atmosphere, both from the University of Michigan: the Global Ionosphere Thermosphere Model (GITM) and the Global Ionosphere Thermosphere Model-2 (GITM2) [22]. They use different explicit upwind schemes for the advection and GITM2 uses implicit time integration for the stiff source terms.

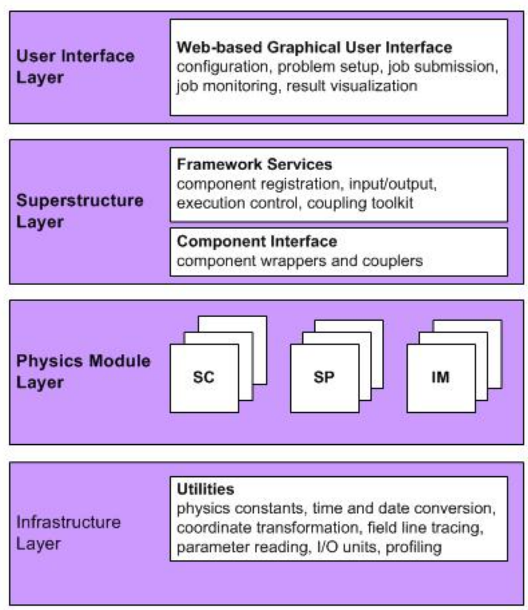


Figure 1 The layered architecture of the SWMF.

3.2 Technical Description

The Space Weather Modeling Framework consists of a User Interface Layer, a Superstructure Layer, a Physics Module Layer, and an Infrastructure Layer, as shown in Figure 1. The Superstructure Layer contains the Component Interface (wrappers and couplers) and the Framework Services. The latter consists of software units (classes) which implement component registration, setting the parallel layout, reading and distributing the input parameters, control of component execution and coupling, and the SWMF parallel coupling toolkit which can be used by component couplers.

The Infrastructure consists of utilities, which define physics constants, transformation between different coordinate systems, time conversion routines, time profiling routines, and other lower level routines. The Infrastructure can be used by the physics modules as well as the Superstructure.

The Physics Module Layer contains the physics modules, which have been integrated into the SWMF, and provide appropriate entry points for the Component Interface.

The SWMF is a single executable framework that can operate in sequential, concurrent, or mixed mode. There is a single driver and a collection of admissible components and couplers defined for the framework. Physically meaningful subsets of the components may be run, or substitutions of data sets for active components. There are multiple sequencing and configuration options possible using the driver.

The main driver of the SWMF has two options for execution: time accurate and steady state mode. In steady state mode the components execute a different number of interactions, and their simulation time is not synchronized. The steady state mode is aimed at an optimal convergence towards an (approximate) steady state solution. In time accurate mode the execution is based on the simulation times. The generality of the execution loop relies upon the presence of case statements that direct the framework to the correct specific run or coupling method.

4 THE CENTER FOR INTEGRATED SPACE WEATHER MODELING (CISM)

Like the SWMF, the CISM framework is used to simulate the region from the Sun's surface to the Earth's upper atmosphere. A CISM model with a set of component versions spanning this domain has not yet been run using the CISM framework, or publicly distributed. An end-to-end application has been run internal to the CISM project using file transfer for some interactions that will eventually be handled using the framework. Many of the component versions within CISM are also models that may be executed standalone.

4.1 Physics Domains

The main components within the CISM framework are the Solar Corona, Solar Wind, Global Magnetosphere, Inner Magnetosphere, and Thermosphere-Ionosphere.

The version of the Solar Corona is the MAS model from SAIC, which uses the magnetic field deduced from observations of the solar surface to provide the lower boundary for the code [23]. To model the Solar Wind from the inner heliosphere to the interaction with the interstellar medium, CISM uses the ENLIL code from the University of Colorado [24,25,26]. The Global Magnetosphere is the Lyon-Fedder-Mobarry (LFM) model [27]. Like SWMF, CISM uses the Rice Convection Model for the Inner Magnetosphere component [18]. The version of the Thermosphere-Ionosphere component is the Thermosphere-Ionosphere Nested Grid (TING) model, a three-dimensional code that has been adapted from the NCAR thermosphere-ionosphere general circulation model [28].

4.2 Technical Description

The CISM group has decided on a strategy that loosely couples codes. Each component is in a separate executable, and the framework runs those executables and transfers data among them. The CISM framework is based on two general tools: InterComm, from the University of Maryland [29], and Overture, which was developed at Los Alamos and is now maintained at Lawrence Livermore National Laboratory [30].

InterComm is a runtime library that achieves direct data transfers between data structures managed by multiple data parallel languages and libraries in different programs. Such programs include those that directly use a low-level message-passing library, such as MPI. Each program does not need to know in advance (i.e. before a data transfer is desired) any information about the program on the other side of the data transfer. All required information for the transfer is computed by InterComm at runtime. Such a data transfer requires that all processes of the sender and receiver programs locate data elements involved in the data transfer and that a mapping be specified between the data elements in the two data structures. Using the data distribution and mapping information, InterComm generates all the information required to execute direct data transfers between the processes in the sender program and the receiver program, and stores the information in a communication schedule. InterComm requires the PVM library.

Overture is an object-oriented framework written in C++ and Fortran for solving problems on a structured grid or a collection of structured grids. In particular, it can use curvilinear grids, adaptive mesh refinement, and the composite overlapping grid method to represent problems involving complex domains with moving components.

CISM uses the InterComm library to send data between components and to sequence component exchanges. Data transformations, for example grid interpolation and physical data translation, are implemented using Overture. The Overture coupling code is isolated from the component versions themselves, either by running it as a separate executable or by calling it as a subroutine from one or both components involved in the data exchange. Unlike ESMF and SWMF, where data is exchanged at the beginning or end of component execution, in the

CISM framework data exchanges may be placed from anywhere in a component to anywhere in another component.

Figure 2 shows a conceptual picture of how two codes can be linked using InterComm and Overture. Model B is a calculation embedded in the larger simulation A. The pseudocode shows that InterComm uses a simple put/get paradigm within components, and that Overture provides a high-level syntax for implementing interpolations and other transformations. A full description of this example is provided in [5].

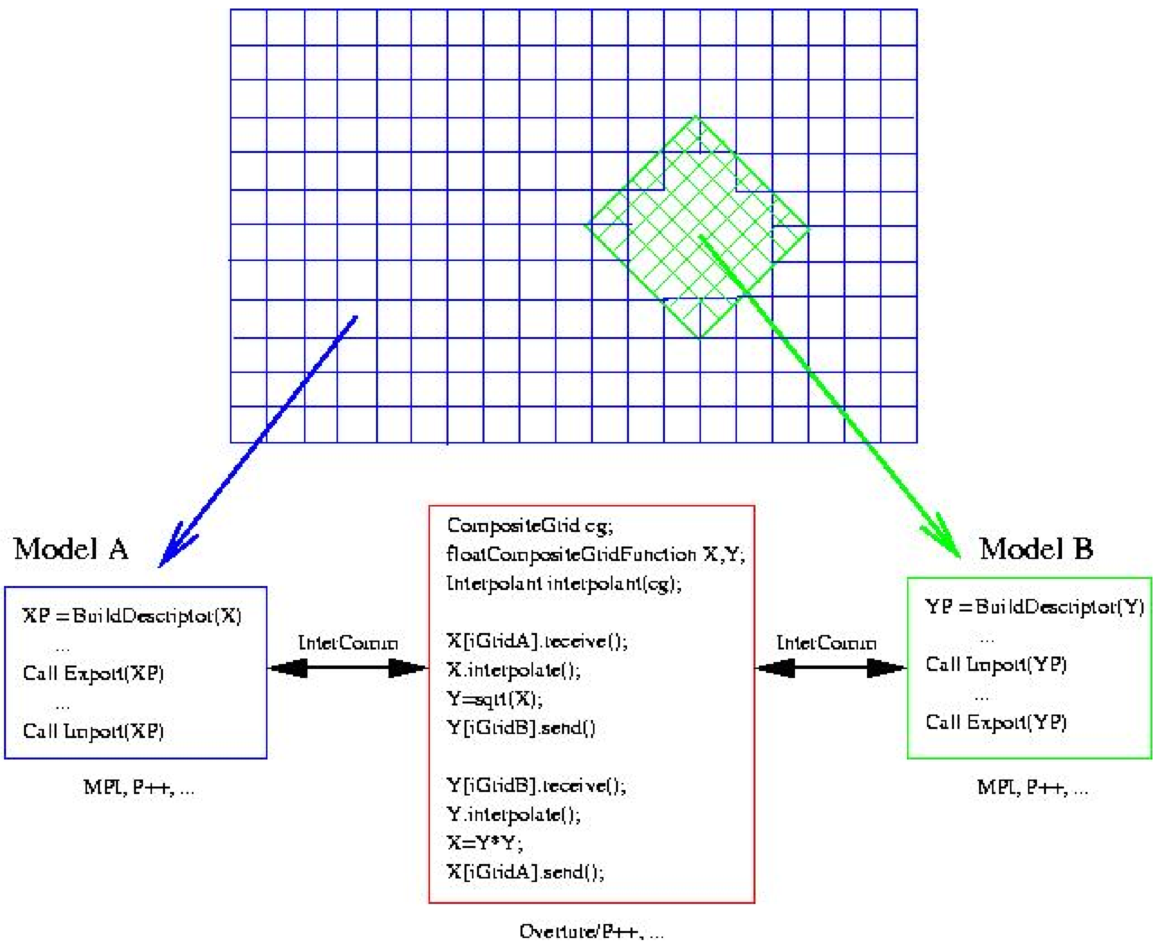


Figure 2 An example showing how Overture and InterComm can be used to couple two simulation codes.

5 THE EARTH SYSTEM MODELING FRAMEWORK (ESMF)

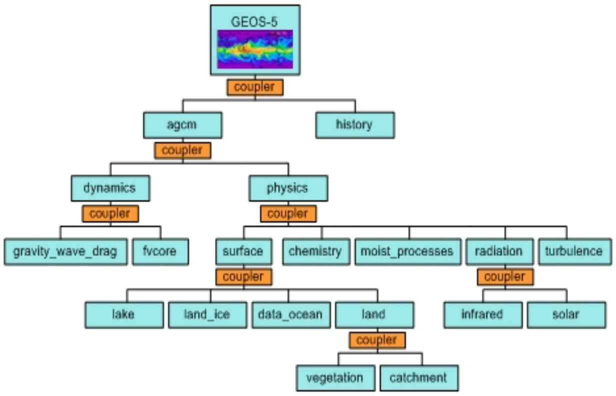
The ESMF project is a large, multi-agency collaboration whose goal is to develop common modeling infrastructure and deploy it in climate, weather, and data assimilation applications. The framework is publicly distributed and has been used to implement and restructure a number of models, including the GEOS-5 atmospheric GCM [31,32], a coupled UCLA atmosphere-ocean model [33], the MITgcm ocean model [34,32], the POP ocean [35], and a coupled atmosphere-data assimilation model at NCEP [32]. These models may have many components (GEOS-5), several (UCLA), or only one (POP), and their current status varies from prototype to production. Groups are funded and in the process of evaluating and using ESMF to couple codes within the DoD Battlespace Environments Institute modeling system [32], the newly funded MAP modeling system for climate variability [32], the CCSM modeling system for climate prediction [36,37,32], and the Flexible Modeling System at GFDL, which also focuses on climate [38,32].

5.1 Physics Domains

Components created using ESMF include atmosphere, ocean, sea ice, land, data assimilation system, atmospheric dynamics, atmospheric physics, and individual physics parameterizations.
ESMF has also been used to create coupler and I/O components. Specific instances of components supported will not be listed here; many are described in the on-line ESMF Component Database [39].

ESMF anticipates supporting a small set of space weather components, including an Inner Heliosphere and a Ionosphere-Plasmasphere. The version of the Inner Heliosphere is the Hakamada-Akasofu-Fry (HAF) Kinematic Solar Wind Model, from the University of Alaska, Fairbanks, and EXPI [40]. The version of the Ionosphere-Plasmasphere is the Global Assimilation of Ionospheric Measurements (GAIM) model developed at Utah State University, University of Colorado, Boulder, and the University of Washington [41].

Figure 3 Structure of the GEOS-5 atmospheric general circulation model.



5.2 Technical Description

Like the SWMF, the ESMF consists of a Superstructure Layer that contains components and couplers and an Infrastructure layer of utilities for parallel communication, regridding tools, time conversion routines, and other functions. There is a Physics Module Layer in between, where user codes must provide appropriate entry points for the Component Interface. The Infrastructure can be used by the physics modules as well as the Superstructure. ESMF does not include a GUI for job configuration and execution, as does the SWMF. Figure 3 illustrates the layered architecture of the framework.

ESMF is a single executable framework that allows components to run sequentially, concurrently, or in a mixed mode. There is limited support for multiple executables, in that a simple test has been developed that allows ESMF components to be created and communicate as separate programs.

ESMF offers two kinds of components: a Gridded Component (GridComp), which is assocatied with a physical domain, and a Coupler Component (CplComp). All Gridded Components and Coupler Components possess initialize, run, and finalize methods with standard interfaces. These methods can be multi-phase.

User code must be split up so that it too contains clear initialize, run, and finalize subroutines that match standard interfaces. The arguments must be ESMF data structures, though these data structures need not be used in the body of the code. Users set entry points within their code so that their initialize, run, and finalize subroutines are callable by the framework.

ESMF components exchange information with other components only through States. A State is a Fortran derived type that can contain other ESMF types representing fields, bundles of fields on the same grid, arrays, or other States. A Gridded Component is associated with an import State and an export State.

ESMF components are arranged in a hierarchical structure to form applications. An example is shown in Figure 4, which describes the architecture of the GEOS-5 AGCM from NASA Goddard. This model was built from the ground up using ESMF. Each box, including the couplers, is an ESMF component with a standard interface. The system can be systematically extended. Exchanges are facilitated both for large, composite components, such as the atmospheric dynamics package, and at the level of a particular physics parameterization.

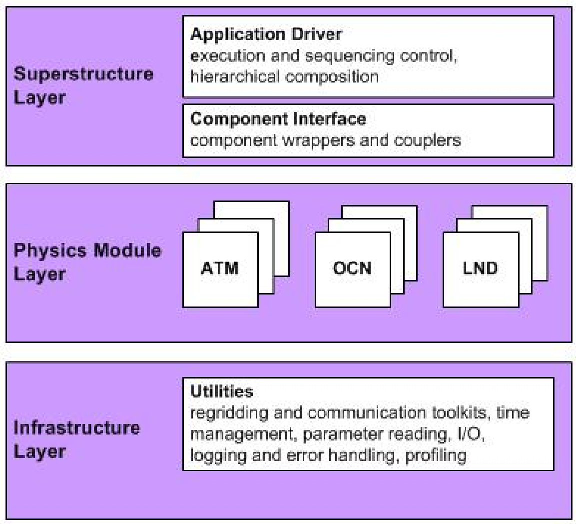


Figure 4 The layered architecture of the ESMF.

6 COMPARISON OF FRAMEWORKS

Table 1 summarizes, in a very rudimentary fashion, the capabilities of these three frameworks. It does not capture crucial characteristics such as ease of use, ease of adoption and extension, the ability to exchange different component versions within the same framework or among different frameworks, or performance. We leave these for future comparison.

Table 1 Comparison of SWMF, CISM and ESMF.

	SWMF	CISM	ESMF
Technical approach			
Mode of execution	Single executable	Multiple executable	Single executable, limited support for multiple executable
Mode of component concurrency	Sequential, concurrent, mixed	Concurrent	Sequential, concurrent, mixed
Mode of inter-component data exchange	Argument list before/after component run	Put/get anywhere	Argument list before/after component run
Standard data structure for inter-component data exchanges	No, although interfaces are generic	Yes	Yes
Nested components through framework	No	No	Yes
Parallelism	Components may use MPI or OpenMP	Components may use MPI or OpenMP.	Components may use MPI or OpenMP
Execution environment	Command line or GUI interface	Command line	Command line
Supported platforms	Linux cluster with NAG, PGF90, and Lahey compilers, OSF1, Altix, SGI 3000, Mac OSX with NAG and XLF.	Linux, AIX, and Solaris.	AIX, SGI IRIX64, OSF1, Linux with Intel, PGI, NAG, Absoft, and Lahey compilers, Mac OSX with XLF and Absoft, Altix, Cray X1
Computational approach			
Supported grids and grid methods	Supported grids and methods: any 1D, 2D, or 3D logically Cartesian grid, and any 2D or 3D block adaptive grid; 2nd order interpolation methods are provided, and user supplied interpolation schemes can also be used.	Any grids and methods supported by Overture.	Intrinsic support only for simple logically rectangular grids, bilinear and first order conservative regrid.

7 PILOT PROJECTS

The first pilot projects to be defined will create interfaces between the ESMF and SWMF frameworks, and between the ESMF and CISM frameworks. If these results are successful, we anticipate the development of a pilot project that will demonstrate interoperation among the CISM, ESMF, and SWMF frameworks.

7.1 SWMF and ESMF Pilot Project

In this pilot project, we will make it possible to run SWMF as an ESMF component that provides input for Earth system models (such as cloud formation). In addition we will provide a modest extension to the ESMF system that will simplify the SWMF/ESMF coupling, and will help to define a paradigm for future collaboration between the two projects. This work is underway.

The specific tasks are the following:

1) We will provide templates to use the ESMF superstructure and access the full functionality of SWMF. Here the primary work involves modifying the control structure of SWMF so that it is ESMF compliant and can be run as a single integrated unit within ESMF. This task has already been completed.

2) We will demonstrate that SWMF and ESMF can exchange field information on a uniform grid. We will add modules that run within SWMF to show that SWMF can produce useful field data on a uniform grid which is passed to an ESMF component. The entire exchange process will be under the control of ESMF. We will also demonstrate that grid information can be passed in the other direction as well. Here the primary difficulty is that ESMF and SWMF use significantly different mechanisms for coupling components and exchanging gridded information. A module running under SWMF and communicating with ESMF components must incorporate both coupling mechanisms.

3) We will provide ESMF with the capability to represent and create a static three dimensional grid and regrid data from that grid. This capability is a step towards the functionality required by the space community, and will simplify the process of transferring SWMF data into the ESMF system for Task 2. Insofar as possible the code developed will be shared intact by both SWMF and ESMF systems, thus setting a precedent for collaborative development. This task is partially complete.

7.2 CISM InterComm and ESMF Pilot Project

Here we will explore the addition of a multiple executable, put/get data transfer paradigm to the ESMF framework using the CISM InterComm tool. The goals will be to create a coupling interface for ESMF and CISM components, and to add flexibility to the ESMF framework. We plan to use InterComm to execute the inter-component data transfers, but to allow ESMF components to do this through interfaces that are ESMF-defined, and accept ESMF_State data structures. This work has not yet begun.

Currently the ESMF framework is limited to coupling components by passing ESMF_State types through component argument lists. There are a number of situations in which this approach poses problems. The first case occurs when the changes made to a component version to run within a framework must be kept to an absolute minimum. The put/get paradigm generally involves changing fewer lines of code within the component version itself. The second case occurs when a coupling must be executed without returning control from the components involved.

In this demonstration, the specific tasks are:

1) We will create ESMF_StatePut() and ESMF_StateGet() commands that will perform a data transformation between ESMF-based components, utilizing the InterComm package to execute the transfer. We will demonstrate the use of these commands in a simple model.
2) We will demonstrate in a simple model the ability to execute a data transformation between an ESMF component and a CISM component using InterComm, with the ESMF component using ESMF interfaces and the CISM component using native InterComm interfaces.

8 DISCUSSION AND CONCLUSIONS

Previous demonstrations in the Earth science domain have shown that it is possible to define interfaces between existing frameworks. Work is underway via two pilot projects to demonstrate that existing Earth and space frameworks can be connected together to form an integrated system for research and forecasting. These pilot projects should be understood to be proof of concept. Each of the frameworks described in this paper is a complex software package, and any combination of them is likely to be a challenging system to build and run for the average researcher. Nonetheless, this exploration is the first step towards more interoperable Earth and space codes, and towards a more collaborative community.

A natural next step is to explore the possibility of component exchanges between the CISM and SWMF frameworks, perhaps mediated by the ESMF interfaces. An examination of the components included in CISM and the SWMF shows that the decomposition of the framework into components is similar, though not exact. The pilot projects will provide useful information about the technical obstacles to interoperability; these may be less challenging than the scientific and social obstacles.

References

[1] Toth, G., Sokolov, I.V., Gombosi, T., DeZeeeuw, D.I., Hansen, K.C., Powell, K.G., Ridley, A.J.., and Stout, Q.E., 2005. The Space Weather Modeling Framework, *Proceedings of ISSS-7*, 26-31.
[2] Toth, G., Sokolov, I.V., Gombosi, T.I., Chesney, D.R., Clauer, C.R., De Zeeuw, D.L., Hansen, K.C., Kane, K.J., Manchester, W.B., Oehmke, R.C., Powell, K.G., Ridley, A.J., Roussev, I.I., Stout, Q.F., Volberg, O., Wolf, R.A., Sazykin, S., Chan, A., Yu, B., Kota, J., 2005. Space Weather Modeling Framework: A New Tool for the Space Science Community, *Journal of Geophysical Research*, submitted.
[3] Collins, N., Theurich, G., DeLuca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C., da Silva, A., 2005. Design and implementation of components in the Earth System Modeling Framework, *International Journal of High Performance Computing Applications*, in press.
[4] Hill, C., DeLuca, C., Balaji, V., Suarez, M., da Silva, A., 2004. The Architecture of the Earth System Modeling Framework, Computing in Science and Engineering 6, No 1, 18-28.
[5] Goodrich, C.C., Sussman, A., Lyon, J.G., Shay, M.A.., Cassack, P.A., 2004. The CISM code coupling strategy, *Journal of Atmospheric and Solar-Terrestrial Physics* 66, 1469-1479.
[6] Fayad, M.E., Johnson, R.E., Schmidt, D.C., Building Application Frameworks: Object-Oriented Foundations of Framework Design, John Wiley and Sons, 1999.
[7] Szypersky, Clemens, Component Software, Addison-Wesley Professional, 2002.
[8] http://www.prism.enes.org
[9] http://www.cca-forum.org
[10] http://www.csm.ornl.gov/~mhy/pubs/jwl-AMS04-CCA-final.pdf
[11] http://www.csem.engin.umich.edu/SWMF
[12] Powell, K. G., P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. De Zeeuw (1999), A solution-adaptive upwind scheme for ideal magnetohydrodynamics, *J. Comput. Phys., 154*(2), 284, doi:10.1006/jcph.1999.6299
[13] Roussev, I. I., T. I. Gombosi, I. V. Sokolov, M. Velli, W. Manchester, D. L. De Zeeuw, P. Liewer, G. Toth and J. Luhmann (2003), A Three-dimensional Model of the Solar Wind Incorporating Solar Magnetogram Observations, *Astrophys. J.*, 595, L57
[14] Gibson, S. E., B. C. Low (1998), A Time-Dependent Three-Dimensional Magnetohydrodynamic Model of the Coronal Mass Ejection, *Astrophys. J., 493*, 460, doi:10.1086/305107
[15] Roussev, I. I., I.V. Sokolov, T.G. Forbes, T.I. Gombosi, M.A. Lee, and J. I. Sakai (2004), A Numerical Model of a Coronal Mass Ejection: Shock Development with Implications for the Acceleration of GeV Protons, *Astrophys. J., 605*, L73
[16] K´ota, J., W. B. Manchester, D. L. De Zeeuw, J. R. Jokipii, and T. I. Gombosi (2005), Acceleration and Transport of Solar Energetic Particles in a Simulated CME Environment,
to be submitted to *Astrophys. J.*
[17] Sokolov, I. V., I. I. Roussev, T. I. Gombosi, M. A. Lee, J. K´ota, T. G. Forbes, W. B. Manchester and J. I. Sakai (2004),A New Field Line Advection Model for Solar Particle Acceleration *Astrophys. J., 616:L171*, L174.
[18] Wolf, R. A., M. Harel, R.W. Spiro, G.-H. Voigt, P. H. Reiff, and C. K. Chen (1982), Computer simulation of inner magnetospheric dynamics for the magnetic storm of July 29, 1977, *J. Geophys. Res.*, 87, 5949
[19] De Zeeuw, D. L., S. Sazykin, R. A. Wolf, T. I. Gombosi, A. J. Ridley, G. T´oth (2004), Coupling of a Global MHD Code and an Inner Magnetosphere Model: Initial Results, *J. Geophys. Res., 109*, 12219 doi:10.1029/2003JA010366
[20] Ridley, A. J., T. I. Gombosi, and D. L. De Zeeuw (2004), Ionospheric control of the magnetospheric configuration: Conductance *Ann. Geophys., 22*, 567
[21] Ridley, A. J. and M. W. Liemohn (2002), A model derived stormtime asymmetric ring current driven electric field description, *J. Geophys. Res., 107(A8)*, 1290, doi:10.1029/2001JA000051
[22] Ridley, A. J., Y. Deng, G. T´oth (2005), A general ionospheric-thermospheric model (GITM), *in preparation*
[23] Mikic, Z., Linker, J.A., Schnack, D.D., Lionello, L.R., 1999. Magnetohydrodynamic Modeling of the global solar corona. *Physics of Plasmas* 6, 2217-2224.
[24] Odstrcil, D., Dryer, M., Smith, Z., 1996. Propagation of an interplanetary shock along the heliospheric plasma sheet. *Journal of Geophysical Research* 101, 19, 973-19,986.
[25] Odstrcil, D., Pizzo, V.J., 1999a. Three-dimensional propagation of coronal mass ejections(CMEs) in a structured solar wind flow 1, CME launched with the streamer belt. *Journal of Geophysical Research* 104, 483-492.
[26] Odstrcil, D. Pizzo, V.J., 1999b. Distortion of an interplanetary magnetic field by three-dimensional propagation of CMEs in a structured solar wind. *Journal of Geophysical Research* 104, 28,225-28,239.
[27] Fedder, J.A., Slinker, S.P., Lyon, J.G., Elphinstone, R.D., 1995b. Global numerical simulation of the growth phase of and the expansion onset for substorm observed by Viking. *Journal of Geophysical Research* 100, 19083.
[28] Wang, W., Killeen, T.L., Burns, A.G., Roble, R.G., 1999. A high-resolution, three-dimensional, time-dependent, nested grid model of the coupled thermosphere-ionosphere. *Journal of Atmospheric and Solar-Terrestrial Physics* 61, 385-397.
[29] http://www.cs.umd.edu/projects/hpsl/chaos/
[30] http://www.llnl.gov/CASC/Overture
[31] http://gmao.gsfc.nasa.gov/systems/geos5/

[32] http://www.esmf.ucar.edu/main_site/meeting_summaries/mtg_0507_commmtg.html
[33] http://www.scd.ucar.edu/dir/CAS2K5/speakers/Mechoso.pdf
[34] http://www.mitgcm.org
[35] http://climate.lanl.gov/Models/POP/
[36] http://www.ccsm.ucar.edu.
[37] Boville, B.A. and P.R. Gent, “The NCAR Climate System Model, Version One,” *J. Climate*, Vol. 11, 1327-1341, 1998.
[38] http://www.gfdl.noaa.gov/~fms
[39] http://www.esmf.ucar.edu, “Applications”
[40] http://www.expi.net/expinet/kinematic.html
[41] http://gaim.cass.usu.edu/index.html