Final Report

From the

National Unified Operational Prediction Capability (NUOPC)

Interim Committee

On

Common Model Architecture (CMA)

1 October 2009

Prepared and Reviewed by:

Common Model Architecture Committee

Scott Sandgathe	University of Washington (Chairman)
Dan Sedlacek	DOD/AF/AFWA
Mark Iredell	NOAA/NWS/NCEP/EMC
Thomas L. Black	NOAA/NWS/NCEP/EMC
Thomas B. Henderson	NOAA/OAR/ESRL
Stanley G. Benjamin	NOAA/OAR/ESRL
V. Balaji	NOAA/OAR/GFDL
James D. Doyle	DOD/NAVY/ONR/NRLMRY
Melinda Peng	DOD/NAVY/ONR/NRLMRY
Roger Stocker	DOD/NAVY/CNMOC/FNMOC
Timothy J. Campbell	DOD/NAVY/ONR/NRLSSC
Lars Peter Riishojgaard	JCSDA
Max J. Suarez	NASA/GSFC/GMAO
Cecelia DeLuca	NCAR/ESMF
William Skamarock	NCAR/MMM
William P. O'Connor	CSSS.NET (NUOPC staff)

Submitted by: SCOTT SANDGATHE, CMA Committee Chair

Approved by: FREDERICK TOEPFER, NUOPC Project Manager

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY IV

1.1 OVERVIEW IV
1.2 CMA COMMITTEE PROCESS IV
1.3 KEY RECOMMENDATIONS V
1.4 COST AND SCHEDULE V

2. INTRODUCTION 1

2.1 NUOPC GOALS 1
2.2 CMA GOAL 2

3. TECHNICAL APPROACH 3

3.1 COMMON MODELING INFRASTRUCTURE AND ESMF 3
3.2 ACHIEVING TARGET INTEROPERABILITY WITH THE NUOPC LAYER 4
3.3 SOFTWARE MANAGEMENT 5

4. OBJECTIVES AND RECOMMENDATIONS 7

OBJ1 Technical interoperability 7
OBJ1.1 Component interfaces and behavior 8
OBJ1.2 Component couplers 10
OBJ1.3 Timekeeping 10
OBJ1.4 Grid and data representation 11
OBJ1.5 Metadata and documentation 11
OBJ1.6 Grid transformations 13
OBJ1.7 Data formats and output metadata 14
OBJ1.8 I/O components 15
OBJ1.9 Diagnostics and post-processing 15
OBJ1.10 Configuration files 16
OBJ1.11 Component unit tests 16
OBJ1.12 Portability 16
OBJ2 Scientific interoperability 17
OBJ2.1 Common physical architecture 17
OBJ2.2 Physical constants 17
OBJ3 Ease of ESMF adoption and use 19
OBJ3.1 Maintain source code and documentation 19
OBJ3.2 Update functionality as required 19
OBJ3.3 Limit performance and memory overhead 20
OBJ3.4 Backwards compatibility 20
OBJ3.5 Routine user support 20
OBJ3.6 User training 21
OBJ3.7 Intensive application support 21
OBJ3.8 Compliance checker 21
OBJ4 Code access and distribution 21
OBJ4.1 Configuration management 21
OBJ4.2 Distribution portal 22
OBJ4.3 Security, licensing, and export control 23
OBJ5 Coding standards 23
OBJ6 Management and evolution of code and conventions 24

5. LEGACY VS. FUTURE SYSTEMS 25

6. COST AND IMPLEMENTATION **26**
APPENDIX 1 ..... **29**
APPENDIX II **32**
APPENDIX III **34**
APPENDIX IV **37**

1. Executive Summary

1.1 Overview

This document contains the recommendations of the Interim Committee on Common Model Architecture (CMA) for the National Unified Operational Prediction Capability (NUOPC) for enhancing interoperability and accelerating transition of research into operations at the primary U.S. operational numerical weather prediction (NWP) centers. *These recommendations are not intended to be comprehensive or immutable, but to evolve as technology and requirements evolve*.

These recommendations include:

- Architecture, framework and standards guidelines for legacy and future global numerical weather prediction (GNWP) systems,
- Schedule and cost for implementation, and
- Organizational structures and procedures for developing, adopting, maintaining and monitoring compliance of NUOPC standards.

The goal of these recommendations is to put forward interoperability standards that will facilitate the following NUOPC goals:

- Improve collaboration on development among government agencies,
- Accelerate the transition of new technology into the operational centers, and,
- Implement ways to enhance broad community participation in addressing the National research agenda.

The Committee feels these goals can be met by addressing the following aspects of model architecture:

- Model-to-model interface standards for operations centers.
- Multi-model ensembles for multi-center operations.
- Component reuse.
- Moving new development to operations.
- Moving research to development and/or using development codes in research.

1.2 CMA Committee Process

Initial discussions focused on establishing a common vocabulary and reaching agreement on a set of basic positions, one of which was to use the Earth System Modeling Framework (ESMF) (http://www.esmf.ucar.edu) as the basis for the common model architecture. The next step was to outline the level of interoperability feasible for operational codes (Appendix I). The Committee converted this outline into the table of recommendations, and identified the actions necessary to implement the target level.

Analysis showed that achieving the desired interoperability level required technical development as well as agreement on usage, metadata, and other standards. Progress was made on several of these conventions. Costs were estimated based on the resources required to implement the recommendations. The organizational structure for technical development was the last item discussed. This is an important aspect of this report, since the organization must ensure operational needs are satisfied, engage the broader community, promote good software practices, and enable the infrastructure and applications to evolve. The proposed structure is an extension of the existing ESMF organization. It may require integration with management bodies proposed by other NUOPC Interim Committees.

1.3 Key Recommendations

The Committee developed a set of recommendations in six areas that address both near-term and long-term interoperability goals:

- Technical interoperability including component interfaces and behavior, component couplers, timekeeping, grid and data representation, metadata and documentation, grid transformations, data formats and output metadata, I/O components, diagnostics and post-processing, configuration files, component unit tests, and portability.
- Scientific interoperability, including a common physical architecture and standard treatment of physical constants.
- Functions to improve the usability of ESMF involving maintenance, new capabilities and features, support and training, and software to check compliance with standards.
- Functions to improve code access and distribution including configuration management.
- A minimal set of coding standards to enhance interoperability.
- A management structure to ensure coordinated development and evolution of software and standards.

1.4 Cost and Schedule

The detailed schedules are presented in the report and costs are summarized below in Figure 1. It is anticipated that the initial adoption of the CMA recommendations would be complete by the end of calendar year 2012. The adoption and implementation of the CMA recommendations involve several key actions:

1. The establishment of a NUOPC Content Standards Committee (CSC) charged with reaching agreement on detailed aspects of recommendations, such as standardized physical constants and metadata (Appendices III and

IV). We recommend this activity commence immediately upon acceptance of this report.

2. The completion of a software layer that will enable NUOPC to meet the target level of interoperability. This layer will be implemented by the ESMF Core Development Team in collaboration with operational and other NUOPC stakeholders (Appendices III and IV), and is expected to be completed during FY12. Its development will be coordinated with the activities of the Content Standards Committee.
3. The ESMF release v5, after which ESMF releases will be backwards compatible, is currently in development under funding from DoD, NASA, NOAA, and NSF and is scheduled for delivery by end of calendar year 2010. The completion of this release will help ensure that codes adopting ESMF will not need reworking due to interface changes. It will also free resources for NUOPC-specific developments.
4. The implementation of NUOPC ESMF standards into legacy operational systems at the operational centers. This effort can commence prior to the release of ESMF v5 and the NUOPC software interoperability layer but cannot be completed until their delivery. These actions will begin in FY10; however, the principal effort will be in FY11 and FY12.
5. Links should be created between NUOPC executive management and technical management. These may include reporting requirements for technical groups, cross-memberships on committees, and other strategies.
6. The establishment of a NUOPC Future Model Architecture Committee charged with defining the physical architecture of the next generation of models, the physical processes they contain, and how they are connected to other components (Appendices III and IV)

The primary cost drivers are the development and maintenance of ESMF, the creation of a NUOPC software interoperability layer, and the initial adaptation of legacy software to the NUOPC standards. There is also a smaller ongoing cost to maintain coordination and collaboration through standing management committees and to maintain software in compliance with NUOPC standards. These costs are illustrated below in Figure 1 and detailed in Section 6.

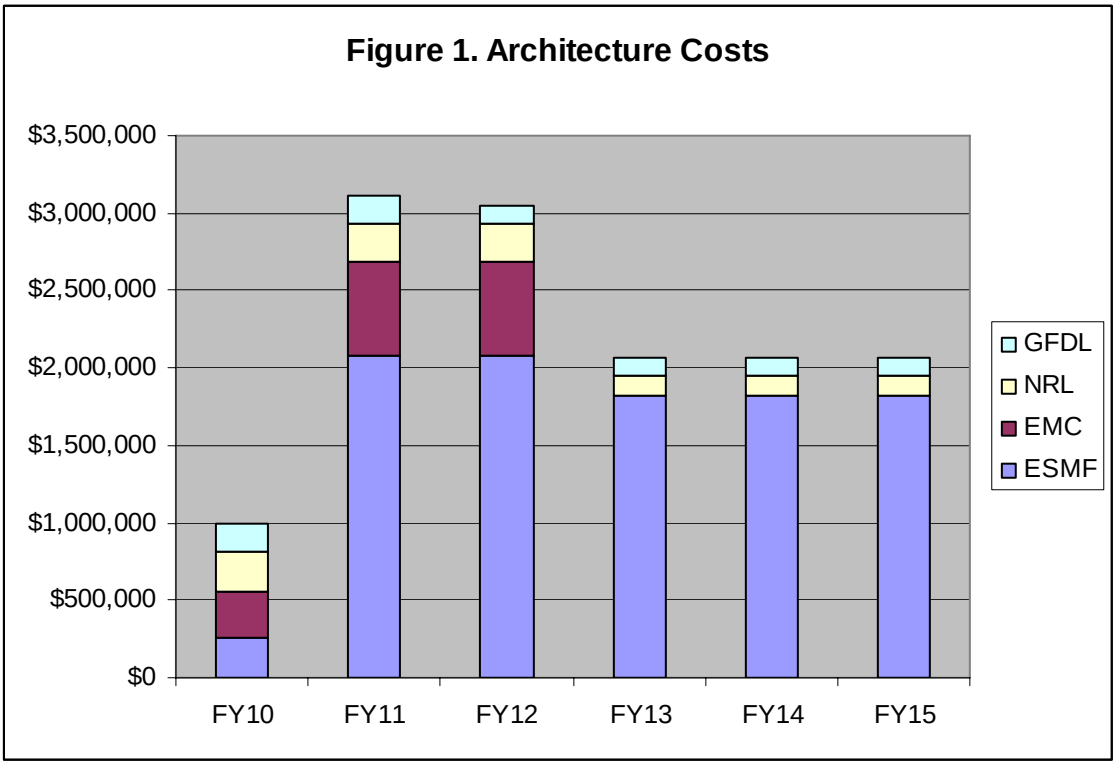
Figure 1. Architecture Costs
$3,500,000
$3,000,000
$2,500,000
$2,000,000
$1,500,000
$1,000,000
$500,000
$0
FY10
FY11
FY12
FY13
FY14
FY15
GFDL
NRL
EMC
ESMF

2. Introduction

This document contains the recommendations of the Interim Committee on Common Model Architecture (CMA) for the National Unified Operational Prediction Capability (NUOPC) for enhancing interoperability and accelerating transition of research into operations at the primary U.S. operational numerical weather prediction (NWP) centers. These recommendations are not intended to be comprehensive or immutable, but to evolve as technology and requirements evolve.

These recommendations include:

- Architecture, framework and standards guidelines for legacy and future global numerical weather prediction (GNWP) systems,
- Schedule and cost for implementation, and
- Organizational structures and procedures for developing, adopting, maintaining and monitoring compliance of NUOPC standards.

In order to aid in comprehension of this document, the following definition of a system architecture is offered:

> The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.
>
> By "externally visible" properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. The intent of this definition is that a software architecture must abstract away some information from the system (otherwise there is no point looking at the architecture, we are simply viewing the entire system) and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction. (*Bass, Clements, and Kazman. Software Architecture in Practice, Addison-Wesley 1997*)

Put more succinctly, a system's architecture is defined by: (1) The components of a system; (2) The functions of the components; and, (3) The relationships and interaction between the components. In this document, the CMA committee puts forth a recommendation for the definition and design of the initial architecture for the NUOPC GNWP capability.

2.1 NUOPC Goals

The overarching goal of NUOPC is to accelerate the rate of improvement in the U.S. National environmental prediction capability, focusing initially on the global model enterprise. Improvements in predictive capability are expected to result in better severe weather warnings (hurricanes, tornadoes, snow storms), better cost avoidance for weather sensitive industries (agriculture, transportation, utilities, defense), and better informed

decision making for industry, defense and the general public. NUOPC proposes to accelerate improvement of our National capability in the following ways: (1) Implementing a global atmospheric ensemble system designed to enhance predictive capability; (2) Clearly articulating operational requirements and articulating a corresponding National research agenda, with initial emphasis on hurricane prediction and ceiling/visibility forecasts; (3) Sharing the predictive burden among the operational agencies; (4) Promoting collaboration on development among government agencies; (5) Accelerating the transition of new technology into the operational centers; and, (6) Implementing ways to enhance broad community participation in addressing the National research agenda. These goals will be aided by developing a common model architecture and other software-related standards. Even the "National research agenda" is facilitated when it can be addressed in terms of a common or collaborative system with interchangeable working parts since all agencies will benefit from individual research efforts. Almost every industry where there are competing or multiple providers is enhanced by adopting standards.

2.2 CMA Goal

The goal of the Common Model Architecture Committee is to recommend interoperability standards that will facilitate the following NUOPC goals:

- Improve collaboration on development among government agencies,
- Accelerate the transition of new technology into the operational centers, and,
- Implement ways to enhance broad community (developmental and climate center, university) participation in addressing the National research agenda.

The Committee feels these goals can be met by addressing the following aspects of model architecture:

- Model-to-model interface standards for operations centers.
    - Link ocean circulation, ocean wave, aerosol, land, ice, estuary, etc., models more easily and reliably.
    - Accelerate the testing, validation, and implementation of outside codes from other operational centers or development organizations.

- Multi-model ensembles for multi-center operations.
    - Enhance interoperability of physics/dynamics suites for creating ensemble diversity and for more rapidly testing new component packages.

- Component reuse.
    - Facilitate the interoperability and reuse of atmospheric physics and dynamics, land, etc., components by standardizing their overall structure and coupling.

- Transition of new development to operations.
    - Accelerate the testing of new developments in an operational environment.
    - Test new physics against operational code.
    - Recalibrate physics suites when new algorithms are added.
    - Assure new code, documentation and testing meets operational center standards.

- Accelerate research transition to operations and/or using development codes in research.
    - Allow research groups to quickly add established models to experimental models such as adding an operationally certified atmospheric model to an experimental ocean model or environmental quality model developed by a research group.
    - Enhance understanding of code through standardized documentation, naming, physical constants, and code structure.
    - Enable research groups using operational models to more readily contribute to improvement of individual code modules, documentation of skill characteristics, and identification of bugs.

3. Technical Approach

3.1 Common Modeling Infrastructure and ESMF

Roughly a decade ago, through a National Research Council report *Improving the Effectiveness of U.S. Climate Modeling, National Achdemies Press, 2001*, and other forums, weather and climate modelers identified common modeling infrastructure as a key element of a more effective U.S. Earth science modeling program. NASA initiated the development of the Earth System Modeling Framework (ESMF) (http://www.esmf.ucar.edu) in 2002 as a realization of the common modeling infrastructure concept. A Core Team was established at NCAR to implement the framework, and a multi-agency team of modelers, drawn from both research and operational centers, contributed to its development and incorporated ESMF into their applications. The design of ESMF was heavily influenced by existing institutional frameworks, particularly the Flexible Modeling System (FMS) at NOAA GFDL and the NASA Goddard Earth Modeling System (GEMS).

A second ESMF development cycle started in 2005, with broadened project sponsorship that included NASA, NOAA, the Department of Defense (DoD), and NSF. The ESMF management structure was amended to include bodies and processes that could support broad participation, mediate amongst numerous stakeholders, and prioritize development tasks in a coherent and transparent fashion.

Now entering its seventh year, ESMF supports a component-based architecture, a variety of coupling configurations, the representation of unstructured and structured grids, and several interpolation options. It includes a set of fully-featured toolkits for parallel communications, calendar and time management, metadata handling, and other common modeling functions. A comprehensive test suite is bundled with the source distribution, and the software is supported on a broad range of platform/compiler combinations. The ESMF team expects to complete the functional requirements laid out in its initial specification by the end of calendar year 2010.

ESMF has been integrated in varying degrees into the NASA Goddard Earth Observing system (GEOS-5) atmospheric general circulation model, the NOAA Global Forecasting System (GFS) and NCEP Environmental Modeling System (NEMS), the Community Climate System Model (CCSM), the Weather Research and Forecasting (WRF) Model, the Hybrid Coordinate Ocean Model (HYCOM), the Coupled Ocean-Atmosphere Mesoscale Prediction System (COAMPS), the Simulating Waves Near Shore model, and numerous others. For the Earth science domain, ESMF is the modeling framework most prevalent in the U.S. at this time.

3.2 Achieving Target Interoperability with the NUOPC Layer

While ESMF tools have proven useful in implementing multi-component modeling systems and coupling individual models, ESMF use has not yet resulted in widespread component interoperability. There are a number of reasons for this:

1. Distribution of ESMF to customers while the framework was in active development has limited interoperability due to gaps in functionality and differences between versions. Completion of the framework and establishment of a backwards compatibility policy for ESMF v5 and later versions will address these issues.
2. Modelers have implemented ESMF at different levels in their applications. For example, some applications contain separate ESMF components representing atmospheric physics and dynamics. Others contain only a single ESMF atmospheric component with embedded physics and dynamics. The CMA Committee has established guidelines for which components should have ESMF

interfaces, though whether and how to implement the atmospheric physics and dynamics split is still an open question for some types of models.

3. The ESMF interfaces offer multiple options for representation of the same quantity, and multiple interpretations of some of their arguments. Although this increases the flexibility of the framework, it makes it difficult to achieve interoperability without the disambiguation provided by additional metadata and usage conventions. The CMA committee addresses this central issue through the creation of a *NUOPC Interoperability Layer* that will specify, through conventions, metadata, and code templates, how NUOPC applications will implement ESMF. While this will not make components within NUOPC "plug-and-play," it will enable NUOPC components to achieve a target level of interoperability. Some or all elements of the NUOPC layer may become part of the main ESMF distribution in order to increase interoperability across a broader community.
4. The ability to exchange different versions of the same component requires an underlying *common physical architecture* – what physical processes each component contains, and how it is connected to other components. The CMA committee has identified the need for describing the physical architecture of the components that will have ESMF interfaces, and a Future Model Architecture (FMA) Committee for defining and resolving these issues. The FMA Committee is described in Appendices III and IV.

The recommendations in this report focus to a large extent on extending the current limits to interoperability as described above.

3.3 Software Management

ESMF has an established community governance structure, described in detail in its Project Plan (http://www.esmf.ucar.edu/management/). Figure 2 shows a diagram of this structure, where each box is a separate body or committee. The ovals show the time scales on which each body operates. The green boxes are either new or contain significant new responsibilities.

The governance structure can be summarized as follows. The ESMF organization is comprised of a *Working Project* and *Executive Management*. The Working Project is defined as the team of customers and developers who collaborate day-to-day to build the ESMF product. The Working Project consists of three parts:

- A line-managed ESMF Core Team responsible for building the ESMF software, including unit and system testing, maintenance, user support, and oversight of a web-based collaboration environment. The ESMF Core Team has been extended to implement software for the NUOPC Interoperability Layer.

- A group of active users called the Joint Specification Team (JST) that interacts with the Core Team and broader community, providing requirements and feedback during design and code reviews.
- A Change Review Board (CRB) that integrates and prioritizes the requirements from multiple users and sponsors, prepares development schedules, and reviews and approves releases.
- A new NUOPC Content Standards Committee (CSC) that addresses metadata, physical constants, documentation standards, and other conventions.

The Working Project is funded, guided, and evaluated by its Executive Management. The Executive Management of ESMF is comprised of several bodies. These are:

- An Executive Board charged with scientific and technical leadership.
- An Advisory Board that reports to and guides the Executive Board.
- An Interagency Working Group (IAWG) that coordinates among sponsors.

Both the Working Project and Executive Management interact with the Earth system modeling and related communities, including the computer science community, the software engineering community, and vendors.

This organizational structure offers the following benefits:

- Priorities for software and content standards development are set according to a formal, regular, and transparent process, by leading stakeholders.
- Software products are discussed and reviewed at multiple stages of development, to ensure that they satisfy customer requirements.
- Software is exhaustively documented and tested, and there are clear policies for user support and user contributions.
- The research and broader community is engaged in development and oversight of the project.
- The development of content standards is an activity led by and focused on NUOPC, but it is also coordinated with ESMF software development and will be available to other ESMF customers.

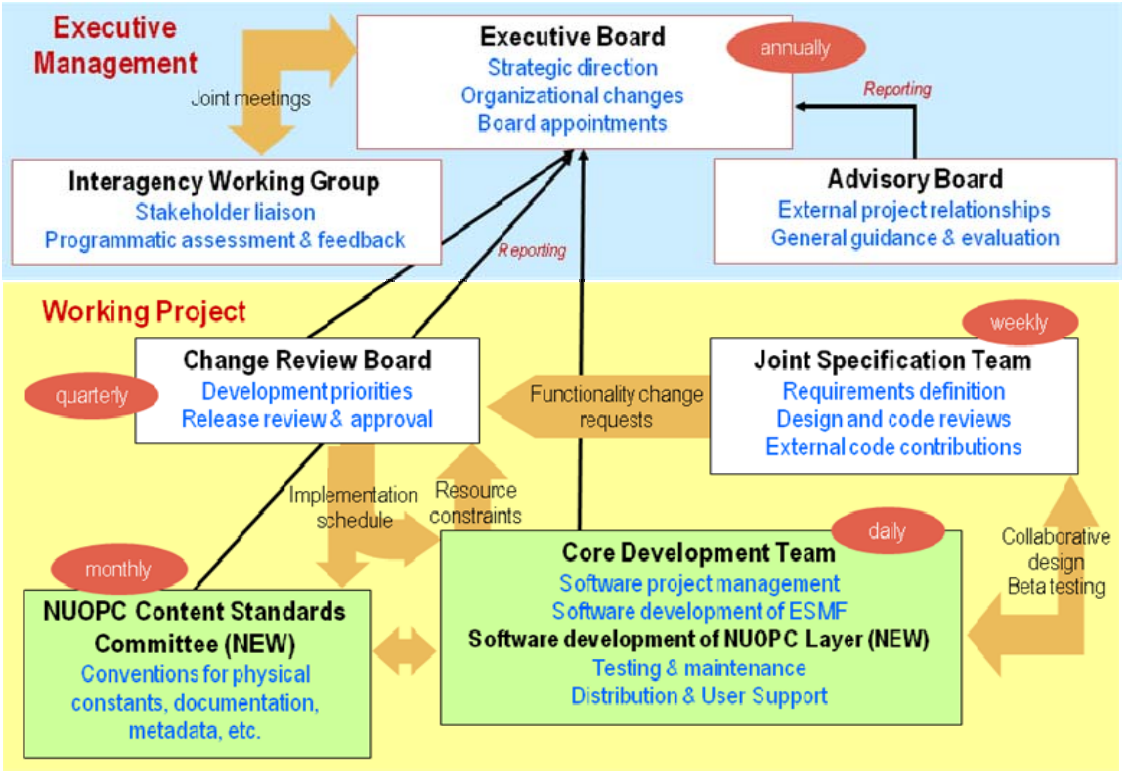

Figure 2. Governance structure of the Earth System Modeling Framework with extensions (green) for the NUOPC program.

4. Objectives and Recommendations

The following are the Committee recommendations for enhancing interoperability and accelerating transition of research into operations at the primary U.S. operational numerical weather prediction (NWP) centers. These recommendations are expected to evolve over time. A table of interoperability specifications is given as Appendix I. The division of labor between committees/groups to implement these recommendations is given in Appendix IV.

OBJ1 Technical interoperability

We expect assembly of new modeling applications from multiple components to require code additions and modifications. However, standard interfaces and tools, usage conventions, and metadata should facilitate interoperability. The level of interoperability desired is outlined below, along with strategies for achieving it.

OBJ1.1 Component interfaces and behavior

Standardized software interfaces and behaviors reduce the time required to exchange software components and assemble new applications.

Software components are functionally distinct elements that can be composed to create software applications within a domain such as Earth system modeling. A standard component interface is a set of routines that is applicable to the components in the domain. For example, many components in models that solve prognostic equations have initialize, run, and finalize phases. Each of these phases may be represented by a routine in the standard interface. Although the arguments in these routines are prescribed, they can be designed with sufficient flexibility to ingest and output many different types of data. Standardized component interfaces as defined here do not prescribe what particular data fields (e.g. temperature) are transferred between components.

Common behaviors are likely to be implemented as *component templates*. These are partially implemented components that can be customized. The templates may include specific calls that are targeted for a function, such as coupling or I/O, and may define the manner or order in which their contents should appear.

REC1.1.1 Adopt the community-based Earth System Modeling Framework (ESMF) as the NUOPC coupling framework at the level of model-to-model (e.g. atmosphere-to-ocean) interactions for legacy and future GNWP systems. Adopt ESMF at the level of atmospheric physics-to-dynamics where feasible and useful for legacy and future GNWP systems. Specifically: NUOPC codes will use ESMF standard interfaces for atmosphere, ocean, sea ice and land components, and where reasonable, for atmospheric dynamics and physics.

Future work: Modify NUOPC applications to use ESMF interfaces.

Rationale: ESMF is the current U.S. coupling framework and is being implemented in DoD, NASA, NOAA and other Government and university modeling systems. There are other coupling frameworks, such as the European Partnership for Research Infrastructure in earth System Modeling (PRISM) which uses the Ocean-Atmosphere-Sea-Ice-Soil (OASIS) coupler; however, the U.S. community has chosen an alternative architecture for coupling. It would require a focused design and development process to significantly increase the compatibility of these frameworks. ESMF has been selected as the one in most common usage in the DoD and NOAA.

REC1.1.2 Establish a convention for data ownership in coupling interactions.

In ESMF, there are two types of components: Gridded Components, which represent physical domains (e.g. sea ice), processes (e.g. radiation), or computational functions (e.g. I/O); and Coupler Components, which transform and transfer data between Gridded Components. Data moving between components is carried in a State object.

Generally, import States are read only by a component, and export States are read only by the outside world. Copying of data from import States to internal States or export States is always allowed. The problem comes when the State data being transferred is so large that too much memory and/or overhead would be taken by copying every time. For instance, an atmospheric model with a 100-species chemistry component cannot be expected to copy the tracer data every time advection is invoked. In this case, one Gridded Component must be considered the owner of the State data and other Gridded Components must be allowed to alter the data. The owner is responsible for creating, initializing, and destroying the data. This State data is still passed through the import and export States but as pointers to the actual data. There must be metadata tags in this State data that identify the owner as well as identifying the outside components with write access and how they are expected to manipulate the data. The Coupler Components should be responsible for checking these tags so that the working Gridded Components do not get overburdened with metadata accounting. The setting of which component owns which data should be flexible enough to be established at run time. Also it must be possible to switch between shared and copied strategies at run time for diagnostic or runtime efficiency.

Future work: Modify NUOPC applications to conform to data ownership conventions.

REC1.1.3 Create software templates for Gridded Components and drivers for use in NUOPC applications. Gridded Component templates are likely to encode whether or not Gridded Components have internal ESMF States, how they store metadata, how the data ownership convention is implemented, and how they handle configuration files.

Future work: Design and implement NUOPC templates for Gridded Components and drivers, as part of the NUOPC Interoperability Layer.

Future work: Modify NUOPC applications to use component and driver templates.

OBJ1.2 Component couplers

REC1.2.1 Create templates for Component Couplers for use in NUOPC applications. Templates may encode aspects of inter-component communication such as whether each Coupler Component passes data in only one direction, or is two-way, reconciliation of ocean and land boundaries, unit conversions, identification of one-time and periodic events, time averaging field data, and validity checking. There may be templates for different types of coupling (e.g., atmospheric physics-to-dynamics, nested regions, etc.)

Future work: Design and implement NUOPC templates for component couplers, as part of the NUOPC Interoperability Layer.

Future work: Modify NUOPC applications to use component coupler templates.

OBJ1.3 Timekeeping

Metadata and conventions for timekeeping enable modelers to understand without code inspection whether components can be coupled together.

REC1.3.1 Establish conventions for the use of Clocks. Clocks are ESMF objects that contain time information including the calendar used, simulation start time, simulation stop time, and time step. They can be associated with multiple Alarms that signal one-time or periodic events, and can be run forwards or backwards.

Preliminary convention: The Clock is passed in the argument list to a Gridded Component method from its parent. The intent of the Clock is input only as far as the time attributes are concerned, but the child component may *add* Alarm attributes to it.

A separate Clock may be attached by the Component to the Gridded Component object itself, and may be created and set by the Component. This is in order to pass time information up to the parent for perhaps some reconciliation step done by the parent. This Clock should not interfere with the Clock being passed in. In particular, if the parent has already created the Clock

in the Gridded Component, there is a danger that setting it would overwrite the parent's master Clock.

Future work: Finalize conventions for the use of Clocks. Implement a simple, efficient mechanism within ESMF to enable time information to be passed from child to parent.

Future work: Modify NUOPC applications to follow timekeeping conventions.

OBJ1.4 Grid and data representation

Native component data to be exchanged are wrapped in a higher level format that includes representation of logically rectangular and unstructured grids, and field metadata such as the field name and units. This improves interoperability by reducing the amount of human intervention required to interpret the data – the format itself carries encoded information about the type of grid, data dimensionality, placement of data points in grid cells, and other information.

REC1.4.1 Represent data at the Field level in inter-component interactions.

ESMF offers the ability to store data in either a general Array index-space object, which does not have any information about the associated grid, or in Field objects that include information about the grid. For NUOPC, import and export State data should be stored in ESMF Field objects, not Arrays. Data may also be stored in FieldBundle objects, which are multiple Fields packaged for efficient collective transfer. In this way, grid information is attached to the States.

Future work: Modify NUOPC applications to follow data representation conventions.

OBJ1.5 Metadata and documentation

Components and data structures include sufficient metadata and documentation to describe the scientific and technical aspects of the software at a high level. In the longer term, components and data structures include sufficiently detailed metadata to automatically perform basic compatibility checks and validation, such as unit compatibility checks.

By documentation we mean all the information associated with a component, presented in print or web media, meant to be read by humans. By *metadata* we mean formally structured information with a controlled vocabulary, embedded in the component or accompanying files and meant to be interpreted by software as well as by humans. Metadata may be used to generate portions of the documentation. The ESMF Attribute class enables metadata to be associated with ESMF Fields, Components, and other data structures.

REC1.5.1 Component documentation includes:

- Version number
- Authors, date and change log
- List of all published papers associated with the component
- Main URL for the component
- Abstract of the purpose of the component
- Description of all the phases of initialize, run, and finalize
- Description of how the component interprets the Clock
- Description of configuration files
- Description of error codes
- Description of resources required
- Description of import and export States
- Brief description of grids used or allowed
- Any caveats or warnings
- Use case or example
- How to validate the component

Rationale: Component documentation is critical to the understanding of complex code. It will be rare that transported model components will be "plug-and-play" and, hence, it is necessary for these components to be well documented in order to:

- Reduce the chance for introducing model errors,
- Hasten understanding and modification of code, and,
- Reduce disruptive calls to the original developer(s).

Notes: Documentation is intended for experienced users who understand the scientific aspects of the code.

REC1.5.2 Component metadata includes:

- Version
- Authors, date, and contact information
- Main URL for the component
- Abstract of the purpose of the component
- Description of import and export States
- Description of grids

Future work: NUOPC intends to adopt the Common Information Model (CIM) being developed by the European Union METAFOR project (http://metaforclimate.eu) in collaboration with U.S. partners. The CIM may need to be supplemented or modified in order to support the GNWP community.

Future work: Modify NUOPC applications to follow component metadata conventions.

REC1.5.3 Establish preliminary Field and FieldBundle metadata conventions. The Fields and FieldBundles in States should be identified with a standard NUOPC name in addition to other metadata needed. This name should be identified in a shared table; if possible, we can use the Climate and Forecast (CF) conventions (see: http://cf-pcmdi.llnl.gov/). The name is associated with standard units. Vector components must also be identified as grid-centric or earth-centric.

Future work: In some cases, such as grid descriptions, the CF conventions are inadequate to describe Fields and FieldBundles. NUOPC intends to adopt the Common Information Model (CIM) being developed by the European Union METAFOR project in collaboration with U.S. partners. The CIM may need to be supplemented or modified in order to support the GNWP community.

Future work: Modify NUOPC applications to follow metadata conventions.

OBJ1.6 Grid transformations

REC1.6.1 Where possible, NUOPC codes will use common ESMF routines for generating and applying weights for interpolations between various map projections and grids. If this is not possible, partners will generate

interpolation weights externally, using a standard format in order to apply them using the framework.

Future work: Establish a standard format for externally generated interpolation weights.

Future work: Modify NUOPC applications to use common routines or conventions for data transformations.

OBJ1.7 Data formats and output metadata

REC1.7.1 Support netCDF4 and GRIB2 data formats. There must be sufficient information in the data format to meet minimal CF requirements. In particular, it should be possible to retrieve the parameter name and units, the date and time and time interval attributes, the model or run from which it came, and grid navigation information. The grid navigation should at least allow the retrieval of latitudes and longitudes for each value.

Rationale: Both of these formats are in very wide use in the climate and weather communities (at both research and operational centers) and it would be very difficult to drop either.

Future work: Design and implement, or adopt, a shared software package that would read and write both netCDF and GRIB efficiently. ('Efficiently' may require parallel or asynchronous or some other kind of I/O.) The software should be able to transfer data to and from a distributed ESMF Field or Field Bundle. This software would take some work to create a prototype, but the major cost would be maintaining it as netCDF and GRIB, the CF and grid component requirements, and the physical I/O systems to which it will be ported, can change over time.

Future work: Modify NUOPC applications to adopt standard data formats and output conventions.

OBJ1.8 I/O components

REC1.8.1 Create shared read/write component templates that use a FieldBundle write interface internally for large-scale asynchronous I/O.

Note: Prototypes exist within the Battlespace Environments Institute (BEI), NASA, and the NCEP Environmental Modeling System (NEMS).

Future work: Design and implement NUOPC templates for I/O, as part of the NUOPC Interoperability Layer.

Future work: Modify NUOPC applications to use I/O templates.

OBJ1.9 Diagnostics and post-processing

REC1.9.1 Create shared diagnostic and post-processing tools and conventions. These may be in the form of component templates.

Note: A prototype for this exists within NEMS – it is currently combined with the I/O component, but should probably be separated out as a stand-alone module. [Note: The TTP & UEO Committees are discussing metrics and post-processing.]

Future work: Design and implement NUOPC templates for diagnostics and post-processing, as part of the NUOPC Interoperability Layer.

Future work: Modify NUOPC applications to use templates for diagnostics and post-processing.

OBJ1.10 Configuration files

REC1.10.1 Explore whether common configuration files are desired or feasible.

Future work: If so, design (or adapt) and implement them. Initial recommendations have focused on elements such as model grid, grid spacing, vertical coordinate, etc.

Future work: Modify NUOPC applications to follow conventions.

OBJ1.11 Component unit tests

REC1.11.1 Accompany each NUOPC model or component with a unit test code to ensure the code is properly installed and functioning.

Rationale: Porting code and components to other hardware and compiler systems often results in subtle component failure that is difficult to detect and can lead to insidious problems and/or very difficult to trace degradation of system performance. A simple test code that ensures the component has compiled properly and is generating results as intended greatly reduces the chance of error.

Future work: Few GNWP components already have suitable unit test modules. A clear understanding of what the tests should include must be formulated, and the tests will have to be designed and developed. It would be useful if they adopted a similar test format.

OBJ1.12 Portability

NUOPC infrastructure and application codes should be readily portable to all the platform/compiler combinations that operational centers are currently using. This list may be expanded to include the platform/compiler combinations used at relevant research facilities. ESMF does some of this already.

REC1.12.1 Establish portability requirements for NUOPC codes. NUOPC may maintain a list of the platform/compiler combinations it will expect infrastructure and application codes to run on.

Future work: Decide on portability requirements for NUOPC codes.

Future work: Modify applications to satisfy portability requirement.

OBJ2 Scientific interoperability

OBJ2.1 Common physical architecture

A common physical architecture will be defined that enables greater interoperability of a specific set of weather prediction components. The *physical architecture* is defined as the scientific scope of components – what processes they include – and the relationship of components to each other in the modeling system. For example, the atmosphere and ocean may be peer components in one modeling system, called and coordinated by the same parent component. In another, the ocean may be invoked by the atmosphere. These differences can cause difficulties when moving components between modeling systems. Factors that influence interoperability include which processes components contain, when they are synchronized, points in the time step at which components are called, and fields exchanged.

REC2.1.1 Establish a Future Model Architecture Committee to define a common physical architecture. Agree on the scientific scope and invocation points of major components. This is expected to be an incremental and exploratory process, which focuses on the largest components first.

Rationale: The interoperability of future modeling systems can be increased by agreeing prior to development to a common physical architecture. This will significantly facilitate understanding of different models, coupling of models, and sharing of technological advances.

Future work: Explore the extent to which it is feasible to standardize component scope and invocation points across NUOPC codes. Assess the impact of non-standardization on interoperability objectives.

Future work: Modify NUOPC applications to conform to desired level of common physical architecture.

OBJ2.2 Physical constants

REC2.2.1 Create a physical constants module to be shared by NUOPC applications. Physical constants utilized in model algorithms will be contained within a common module, use standard names, and use standard values to the extent possible. In order to encourage use of precisely the same values for commonly used constants (such as π or the acceleration due to gravity) all such constants

can be declared and set to the desired values in a single FORTRAN module that will be made available to all users. Three steps may then be taken to incorporate those constants into a user's own code: (1) Insert a FORTRAN USE statement for that module where the user wishes to employ those constants; (2) If only some of the constants are desired then add ONLY to the USE statement and list those constants that the user wants; (3) For all of the constants selected from the module that are already present in the user's code but have different names then the user will also add a 'rename list' to the USE statement in which the selected constants from the module are renamed to the names that are present in the user's code. The user's code will now contain the constants selected from the constants module with no further editing required.

Also, wherever software is unable to use standard values, these should be clearly documented in and out of the code. There should also be a standard invalid floating point value, possibly IEEE standard, which may depend on the field.

Rationale: A common module for physical constants will improve interoperability and also help to avoid undetected model conservation losses from incorrect or slightly different use of constants. Adopting standardized names for common physical constants will also assist in understanding code and reduce unintended errors due to misinterpretations. The standard module should encourage use of standard values by providing developers with ready knowledge of constants in use at operational centers.

Notes: In some situations, it may be difficult to standardize constants due to prior model tuning. In these cases, documentation both in and out of the code should reflect the presence and justification for non-standard constant values. This is both to highlight potential interoperability issues and also to prevent the inadvertent damage from changing the values to standard values.

Future work: Continue to evolve these standards. A NUOPC-maintained register of physical standards should be established and a mechanism for recommending and adjudicating additional standards should be established.

Future work: Modify applications to use a standard module, standard names and standard values for physical constants.

OBJ3 Ease of ESMF adoption and use

OBJ3.1 Maintain source code and documentation

REC3.1.1 Provide resources to maintain ESMF. This includes support for test and development staff to address ESMF bugs and feature requests, to monitor and improve consistency of framework interfaces and behavior, and to provide complete reference documentation. Maintenance also includes continued development of the ESMF regression test suite, and operation of comprehensive nightly regression tests.

The ESMF test suite currently includes thousands of unit tests covering all aspects of the framework, use test cases that demonstrate realistic problems and problem sizes (e.g. in input data, grid size, and processor counts), system tests, and examples. The tests are automated to run nightly on about thirty different platform/compiler combinations. Each day of the week tests different scenarios, such as multi-processor, single processor, and threaded system configurations, and a report is generated for each platform/compiler combination. These tests provide rapid feedback on the robustness of new code, new platforms, and new compilers to developers and customers, and ensure that the framework is highly portable.

Rationale: One of the most significant factors in ease of use is ensuring that the underlying infrastructure is robust and responsive to user needs.

Future work: Ongoing maintenance of ESMF.

OBJ3.2 Update functionality as required

REC3.2.1 Provide resources to develop and update ESMF capabilities in response to application requirements. This covers more significant development activities than the feature requests considered under maintenance.

Future work: Ongoing development of ESMF. This is likely to include the development of additional regridding algorithms, expansion of the unstructured grid library, new I/O capabilities, and extensions to time management and metadata handling libraries.

OBJ3.3 Limit performance and memory overhead

REC3.3.1 Monitor ESMF for performance and memory use and optimize the framework for new computing architectures. ESMF and NUOPC infrastructure should impose no more than a 5% performance penalty in time to solution compared to native code.

OBJ3.4 Backwards compatibility

REC3.4.1 After 2010 ESMF interfaces should be backwards compatible following the ESMF 5.0 public release for an indefinite period of time. A process should be established for exceptions to this policy.

REC3.4.2 NUOPC participants should migrate to new ESMF public releases within a year of their release.

REC3.4.3 NUOPC participants will abide by the ESMF release support policy, which is as follows:

"This policy applies to a period of active development, where public releases are produced on a roughly annual basis.

The ESMF Core Team supports the last two public releases (not counting patch increments) of the ESMF software, with the additional constraint that each public release is supported for a minimum of one and a maximum of three years. User support includes responses to requests for new features, bug fixes, ports to new platforms, patch releases, and help understanding and using the software.

Requests related to supported and unsupported versions of ESMF will be prioritized by the Change Review Board and Core Team following the procedure described in the User Support Policy. Requests related to unsupported versions will be processed as time and resources permit."

OBJ3.5 Routine user support

REC3.5.1 NUOPC should maintain an ongoing support function for installation, design, and implementation questions concerning ESMF and the NUOPC infrastructure. Routine user support does not include running NUOPC

applications or implementing ESMF within user codes. This is the level of support that the ESMF team has provided to their users to date.

OBJ3.6 User training

REC3.6.1 ESMF should provide tutorials and higher-level documentation that describe how to use the framework for new users. Materials should include examples, sample applications, demonstrations, an extensive website, and an expanded users' guide.

OBJ3.7 Intensive application support

REC3.7.1 **Future work:** Create an application support team consisting of domain experts who can work with NUOPC participants to implement ESMF and NUOPC infrastructure within their codes. The members of this team will directly assist NUOPC modelers with implementation of recommendations. They are expected to run application codes to and work with modelers for extended periods. These are not functions covered by routine user support.

OBJ3.8 Compliance checker

REC3.8.1 A generic component can be created into which you can insert a component and run an ESMF 'smoke test'. This would be the minimum test to show that the component is ESMF compatible (that the component is installed correctly). This kind of driver is being developed by NCAR for ESMF for their 2010 product completion. This will check compliance at some level automatically. It will check to see if a component does nothing when told to do nothing. It will try to check for correct metadata.

OBJ4 Code access and distribution

OBJ4.1 Configuration management

Access to development versions at other institutions, controlled versioning.

REC4.1.1 Standardize configuration management software used for maintaining and developing NUOPC component and infrastructure code and documentation. Subversion (http://subversion.tigris.org/) is the current preferred configuration management software. All centers either use or are going to Subversion with the exception of NASA. NCEP, GFDL, ESRL, FNMOC and NRL currently

use Subversion. NCAR ESMF and AFWA have conversion to Subversion in their long range plans.

Rationale: To reduce training, increase understanding and ease of use, to ensure all code is maintained to the same standards.

Future work: Infrastructure and applications not using Subversion migrate to Subversion.

REC4.1.2 All code components should be accessible through a common interface such as a single web page or web interface; however, not necessarily a single repository. This may require developing a plan for repository linkage and coordination.

Rationale: There was a strong desire by Committee members to have developers responsible for configuration management of their own code components; however, users should have easy, standardized, access to all authorized components.

REC4.1.3 The configuration management structure should provide for multiple levels of access for both reading and modifying component code.

Rationale: The software developers should have absolute control of the main (trunk) version of component code and when/if to accept changes tested under branch versions. Additionally, researchers may wish to develop and control access to branch versions of component code.

Note: For purposes of intellectual property rights or information assurance, some portions of component code may not be widely releasable. These components; however, may be releasable to smaller groups (i.e., U.S. citizens, individuals signing non-disclosure agreements, etc.).

OBJ4.2 Distribution portal

REC4.2.1 Offer NUOPC templates and codes via a distribution portal. Such a portal would contain released versions of the source code, with metadata that describes the source code. It would not have a configuration management system or repository. The distribution portal is a way for people to see what components are available, understand what the differences are between versions of the codes, and to download the source. Its function is closely

related to data distribution, since data may be offered along with code. Levels of authentication may be available.

Future work: Decide whether portal access (in addition to repository access) is desired.

OBJ4.3 Security, licensing, and export control

REC4.3.1 The configuration management system, the distribution portal, and the ESMF implementation should follow clear standards and guidelines specifically established for NUOPC. Specifically, it should: 1) provide for the prevention of malicious or accidental modification of code; 2) protect the intellectual property rights and/or copyrights associated with the code; and, 3) protect against violation of export control and information assurance regulations.

Future work: Determine what is needed in this area.

OBJ5 Coding standards

It was determined that all organizations have very detailed coding standards and NUOPC would be ill-advised to try to create its own standards. However, there are certain coding practices that would greatly enhance interoperability. These the Committee felt <u>should</u> be adopted:

REC5.1.1 The community appears to be divided on the direction of the vertical indices ('k') used within models. Some models define the k=1 level at the model top, others at the lowest model level. Commonly used physical parameterization components within these models are often written to be consistent with the model selection of the vertical index orientation. We recommend a generalized vertical index within model components such that either vertical index direction can be utilized. For some legacy algorithms and subroutines that may be too complex to adapt to a generalized index, we recommend that the input and output arrays are re-ordered to allow for the generalized vertical index functionality.

REC5.1.2 Global data statements: The use of common blocks, global data statements or other coding methods that result in variables being passed between coupled components via shared memory or other non-obvious manner should be avoided. Global data statements are permitted below the ESMF coupling level as long as they are private within the component (e.g., within an ocean model). If this is impractical for certain legacy codes or for significant model run time

issues, then the use should be clearly documented in and out of the code. Care must be taken with common blocks when running ensembles, such as ensuring they run off different nodes.

Future work: Modify applications to conform to coding standards

OBJ6 Management and evolution of code and conventions

The evolution of software and conventions for NUOPC is fundamentally a technical activity that requires software engineering structures and processes to be effective. The elements of the organization recommended here are intended to encourage best software development practices, to give NUOPC stakeholders direct input into development priorities and schedules, to provide transparency in decision-making, and to support engagement in the development process by a broader community.

The technical organization proposed for NUOPC software standards development is based on the established ESMF management structure, and outlined in section 3.3. The modified organization must be approved by the ESMF Executive Board before implementation.

REC6.1.1 **Future work**: Extend the ESMF Core Team so that it is responsible for software development and application support for the NUOPC Interoperability Layer. The Core Team will receive direction on priorities from a Change Review Board that includes representation from operations centers. The NUOPC Layer is expected to be distinct from ESMF main distribution. The ESMF Core Team is described in Appendices III and IV.

REC6.1.2 **Future work**: Create a NUOPC Content Standards Committee (CSC) with the responsibility for developing conventions for metadata, documentation, physical constants and other areas that require mutually agreed on values and formats. The Standards Committee will not be responsible for the design of software or software interfaces. However, its work will need to be coordinated with the ESMF Core Team, since software will be required to support implementation of some of the conventions. The Chair of the Content Standards Committee is expected to be a member of the NUOPC program staff. The Standards Committee will receive guidance on priorities from a Change Review Board. The CSC is described in Appendices III and IV.

REC6.1.3 **Future work**: Establish a Future Model Architecture Committee to investigate/coordinate what physical processes future model components

contain, and how they are connected to other components. The FMA is described in Appendices III and IV.

REC6.1.4 **Future work:** Creation of a broad forum to encourage communication amongst NUOPC modeling groups, and input from these groups into the infrastructure development process. This group will be encouraged to submit feature requests and requirements, participate in design and implementation reviews, contribute code, and engage in beta testing and joint development of NUOPC software. This group is expected to be a reformulation of and replacement for the ESMF Joint Specification Team, which currently has more than 500 subscribers to its mailing list. Collaboration strategies used in the forum may include meetings, websites, teleconferences, desktop-sharing software such as Webex, wikis, forums, mailing lists, trackers, and software and data catalogues.

5. Legacy vs. Future Systems

The value of modifying legacy systems to conform to common physical and scientific architectures has not been demonstrated in a true cost - benefit analysis. However, the above recommendations should be implemented wherever feasible as they enhance interoperability and understanding.

There are several future GNWP systems under development or being considered, both for weather and for seasonal and climate prediction. There is still considerable debate on the most appropriate physical/scientific structure for these systems as it is likely that multiple dynamic cores will be required to accommodate all users and purposes. These systems will also be machine intensive and require adaptation to specific hardware architectures for optimum performance.

In order to obtain full benefit from U.S. research and development efforts, future GNWP systems should adhere to all NUOPC guidelines including physical architectures to the extent possible. Where hardware or scientific requirements preclude this, all consideration should be made for updating or adjusting NUOPC guidelines.

It is impossible to determine in advance scientific and hardware requirements for future GNWP systems. However, achieving greater interoperability lies mainly in the intelligent and collaborative development of future modeling systems. NUOPC and associated technical bodies will be responsible for increasing interoperability through coordination and evolution of future architecture standards.

6. Cost and Implementation

The following are rough milestones and tasks for implementing the common model architecture as given by participating organizations. All resources (FTE) listed are additional resources required above current resources. Table 1 summarizes all costs by fiscal year and by performing activity.

Table 1. CMA Cost Summary

	FY10	FY11	FY12	FY13	FY14	FY15
ESMF	$ 260K	$ 2080K	$ 2080K	$ 1820K	$ 1820K	$ 1820K
EMC	$ 300K	$ 600K	$ 600K			
NRL	$ 250K	$ 250K	$ 250K	$ 125K	$ 125K	$ 125K
GFDL	$ 180K	$ 180K	$ 120K	$ 120K	$ 120K	$ 120K

NCEP

GFS NEMS Development Tasks

Year 1 (FY10)

- Make NEMS global atmospheric model a full NUOPC component.
- Include FIM dynamics in NEMS.
- Begin coupling of MOM4 and HYCOM ocean models to GFS NEMS.
- Continue coupling of aerosol component to GFS NEMS.
- Complete inclusion of parallel GRIB2 output.
- Complete inclusion of ensemble capability with stochastic forcing.

Year 2 (FY11)

- Complete coupling of MOM4 and HYCOM ocean model to GFS NEMS.
- Complete coupling of aerosol component to GFS NEMS

Year 3 (FY12)

- Complete full (1-way, 2-way, moving) nesting capability from global to storm scale with atmosphere and ocean models.

This effort will take 5 FTE to meet these deliverables in FY12. No additional FTE above current levels is required beyond FY12.

Navy (NRLMRY)

Year 1 (FY10) – Implementation of ESMF in NOGAPS

- Superstructure for NOGAPS.

- Break down NOGAPS into dynamics and physics components
- Include land – surface model
- Complete first version of code
- 2 FTE ($500K)

Year 2 (FY11) – Implementation of ESMF in UM

- Superstructure for UM to include UM in the multi-model ensemble
- No breakdown of UM into dynamics and physics components
- Complete first version of the code
- 1 FTE ($250K)

Year 3 (FY12) and beyond

- 0.5 FTE/year for maintenance

OAR

GFDL IPlan

Deliver MOM4.1 ocean model as a NUOPC component

- Work nearly done
- 0.1 FTE/year continuing maintenance

Deliver atmospheric dynamics and atmospheric physics as separate components complying with NUOPC and NEMS architecture

- Involves 3D coupling
- May require recoding the vertical coordinate
- 1 FTE for 1 year initial development
- 0.25 FTE/year for continued maintenance

GSD

FIM – NEMS development tasks

- Assume NEMS and NUOPC will merge at some point
- 1 FTE/year (current NEMS resources)
- Funding from existing NOAA HPCC budget already directed toward FIM-NEMS development

NCAR ESMF

Year 1 (FY10) – Set up organizational infrastructure, broad system prototyping

- Finalize and implement organizational plan – including reporting, management and staffing for distributed development and support teams, and funding vehicles.
- Set up joint website, trackers, lists, and other communication and management infrastructure, initial code distribution infrastructure, and initial repository access and policies.

- Prototype the component template and highest level (including atmosphere-ocean) coupler template, document them, and distribute them via the web. This activity must address aspects of the common physical architecture. [Implement templates in at least one NUOPC code.]
- Examine relationship of NUOPC templates to MAPL and develop interoperability plan.
- Other code and convention development activities as prioritized by the Operations Review Board and ESMF Change Review Board.

Year 2 (FY11) – Delivery of first production elements

- Migrate ESMF to subversion.
- Assess and evolve NUOPC-wide code distribution and repository strategy.
- Finalize development of the component and highest-level coupler templates and distribute. [Implement in multiple NUOPC codes.]
- Prototype diagnostics, postprocessing and IO templates and distribute. [Implement in at least one NUOPC code.]
- Refine and distribute common physical constants module.
- Finalize component, field, and grid metadata packages. [Implement as ESMF Attributes in at least one NUOPC code.]
- Develop initial conventions for configuration files, working closely with GFDL, AFWA, Navy, etc. [Implement in at least one NUOPC code.]
- Other code and convention development activities as prioritized by the Operations Review Board and ESMF Change Review Board.

Year 3 (FY12) – Assess and refine NUOPC layer

- Finalize development of diagnostics, postprocessing, and IO templates and distribute. [Implement in multiple NUOPC codes.]
- Prototype atmospheric physics-dynamics template and distribute. [Implement in at least one NUOPC code.]
- Refine conventions for configuration files. [Implement in at least one NUOPC code.]
- Clean up documentation and prepare training materials.
- Other code and convention development activities as prioritized by the Operations Review Board and ESMF Change Review Board.

Year 4 and beyond – Maintain and evolve ESMF and NUOPC software and standards.

APPENDIX I
Interoperability Specification Document (DRAFT)

Developed for National Unified Operational Prediction Capability (NUOPC) Common Modeling Architecture Committee
January 12, 2009

Purpose: This document describes the level of interoperability desired by NUOPC partners who are constructing multi-component and multi-model applications. The choices here motivate the adoption of software tools and conventions to facilitate interoperability. Since increased interoperability is associated with increased code preparation effort, the choices here also reflect tradeoffs between interoperability, time, and cost.

Option Table: The table below divides the construction of applications into a set of categories (component interface, timekeeping, etc.). These categories are not orthogonal, but each illustrates a different perspective on interoperability. Every category is associated with a numbered sequence of options that describes steps towards greater interoperability. Levels defined for the various categories are independent – i.e., selecting level 2 across all categories does not represent a consistent approach. Table 2 indicates the desired options and timeframe for achieving those options.

Table 2. Interoperability Goals

Category/Level of Interoperability	**Desired?**		
	Before 2010	2010-2015	Not before 2015/Never
AA Application assembly			
1. Assembly of applications requires code additions and modifications. Components have *ad hoc* interfaces and coupling strategies.			
2. Assembly of applications requires code additions and modifications, but standard component interfaces and tools, usage conventions, and metadata facilitate understanding and assembly.	x	x	
3. Components are "plug-and-play" technically. Application assembly requires XML or GUI specification, not code manipulation.			
4. Components are "plug and play" both technically and scientifically. Preliminary scientific compatibility			

checking and validation is done automatically.			
CI Component interface			
1. A model component is not well-defined.			
2. Components are clearly defined software modules for purposes of reuse and exchange.			
3. Components have a standard calling interface to facilitate generic drivers and communication protocols. Standardization does not include specification of what fields are actually in the import and export state.	x	x	
CC Component couplers			
1. Couplers are written by application developers in an *ad hoc* manner.			
2. Couplers are written by application developers using standard interfaces and tools.	x		
3. Standard couplers can be generated automatically or correct behavior can be deduced at runtime and implemented using generic couplers.		x	
4. Couplers are automatically generated and perform internal compatibility checking or semantic reasoning.			
TK Timekeeping			
1. Component timekeeping is *ad hoc*.			
2. Components use a common timekeeping package to interface with other components.			
3. Metadata and conventions for timekeeping enable modelers to understand without code inspection whether components can be coupled together.	x	x	
4. Applications can be assembled automatically, taking timekeeping into account.			
GD Grid and data representation			
1. No common data format is defined for component variables to be exchanged.			
2. Native component data to be exchanged is wrapped in a basic multi-dimensional array or vector format for inter-component data exchanges.			
3. Native component data to be exchanged is wrapped in a higher level format that includes representation of logically rectangular and unstructured grids and metadata.	x	x	
MD Metadata			
1. Components and data structures (fields, grids, etc.) do not have standard metadata.			

2. Components and data structures include sufficient metadata to describe the scientific and technical aspects of the software at a high level.	x		
3. Components include sufficiently detailed metadata to perform basic compatibility checks and validation, such as unit compatibility checks.		x	
4. Components and data structures include sufficiently detailed technical and scientific metadata to generate component couplings automatically, and to be used in analysis and visualization workflows.			
GT Grid transformations			
1. Interpolation weights can be generated external to the framework.			
2. Interpolation weights can be generated within the framework, based on standard grid and data representations. Externally generated interpolation weights are still allowable.	x	x	
IO IO			
1. No standard IO formats.			
2. Shared read/write components or component templates use a bundle interface internally for IO.		x	
3. Everyone writes CF-compliant netCDF files.			
DG Diagnostics and post-processing			
1. No standard diagnostics.			
2. Shared diagnostic and post-processing tools and conventions. These may be component templates that people could share.		x	
CF Run-time configuration and control files - deferred			
1. No standard format or approach for configuration files.			
2. Documentation of configuration files.		x	
3. Agreed on approach or tool but no standard structure.			
CPA Common physical architecture			
1. No agreement on component scope and invocation points or specific fields exchanged			
2. Agreement on component scope and invocation points		x	
3. Agreement on component scope, invocation points, and fields exchanged			

APPENDIX II
ACRONYMS

AF	Air Force
AFWA	Air Force Weather Agency
BEI	Battlespace Environments Institute
CCSM	Community Climate System Model
CF	Climate and Forecast (metadata conventions)
CIM	Common Information Model
CMA	Common Model Architecture Committee
CNMOC	Commander Naval Meteorology and Oceanography Command
COAMPS	Coupled Ocean-Atmosphere Mesoscale Prediction System
CRB	Change Review Board
CSC	Content Standards Committee
CSSS.NET	Client Server Software Solutions Corporation
DoD	Department of Defense
EMC	Environmental Modeling Center (NOAA NWS)
ESMF	Earth System Modeling Framework
ESRL	Earth Systems Research Laboratory (NOAA OAR)
FMA	Future Model Architecture Committee
FMS	Flexible Modeling System (NOAA GFDL)
FNMOC	Fleet Numerical Meteorology and Oceanography Center (Navy CNMO
FTE	Full Time Employee
GEMS	Goddard Earth Modeling System
GEOS-5	Goddard Earth Observing System (NASA)
GFDL	Geophysical Fluid Dynamics Laboratory (NOAA OAR)
GMAO	Global Modeling and Assimilation Office (NASA)
GNWP	Global Numerical Weather Prediction
GRIB2	Gridded in Binary (version 2)
GSFC	Goddard Space Flight Center (NASA)
HYCOM	Hybrid Coordinate Ocean Model
IAWG	Interagency Working Group
IEEE	Institute of Electrical and Electronic Engineers, Inc.
I/O	Input/Output
JCSDA	Joint Center for Satellite Data Assimilation
JST	Joint Specification Team
MAPL	Model Analysis and Prediction Layer
METAFOR	Common Metadata for Climate Modeling Digital Repositories

MMM	Mesoscale and Microscale Meteorology (NCAR)
MOM4	Modular Ocean Model 4
NASA	National Aeronautics and Space Administration
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction (NOAA NWS)
NEMS	NCEP Environmental Modeling System
netCDF4	net Common Data Format version 4
NOAA	National Oceanic and Atmospheric Administration
NRL	Naval Research Laboratory
NRLMRY	Naval Research Laboratory – Monterey
NRLSSC	Naval Research Laboratory – Stennis Space Center
NSF	National Science Foundation
NUOPC	National Unified Operational Prediction Capability
NWP	Numerical Weather Prediction
NWS	National Weather Service (NOAA)
OAR	Ocean and Atmospheric Research (NOAA)
OASIS	Ocean-Atmosphere-Sea-Ice-Soil (model coupler)
ONR	Office of Naval Research (Navy)
PRISM	Partnership for Research Infrastructure in earth System Modeling
TTP	Technical Transition Processes Committee
UEO	Unified Ensemble Operations Committee
URL	Universal Resource Locator
WRF	Weather Research and Forecasting model
XML	Extensible Markup Language

APPENDIX III
COMMON MODEL ARCHITECTURE COMMITTEE CHARTERS

The following charters are preliminary.

Content Standards Committee Charter

The NUOPC Content Standards Committee (CSC) is the primary committee responsible for evolving and expanding the NUOPC interoperability standards. The CSC will have responsibility for developing conventions for metadata, documentation, physical constants and other areas that require mutually agreed on values and formats. The CSC will not be responsible for the design of software or software interfaces. However, its work will need to be coordinated with the ESMF Core Development Team, since software will be required to support implementation of some of the conventions. It is expected that the CSC will work in close collaboration with the NUOPC Core Development Team (CDT). The CSC will decide model content standards, with input from the CDT, and then give the work to the CDT. For this reason, there should be some individuals who are members of both committees.

The CSC will receive guidance on priorities from the NUOPC PM and the ESMF Change Review Board and pass standards to the CDT for implementation in the NUOPC ESMF layer. It will also interact with the Future Model Architecture Committee to evolve standards suitable for future systems.

The member of the NUOPC staff responsible for the Common Model Architecture will also chair the CSC. The Chair of the CSC should be an ex officio member of the ESMF change review board. This committee will be composed of members from the NUOPC participating operational centers, from the developmental organizations of the Tri-agencies, from other U.S. government agencies (NASA, DOE and FAA), NCAR ESMF and MMM, and, as necessary, other university groups.

The CSC will meet as required to complete definition of the initial NUOPC standards and then at least quarterly thereafter, by teleconference and/or internet with periodic face-to-face meetings.

Core Development Team Charter

The ESMF Core Development Team (CDT) will be responsible for implementing the NUOPC Layer above the present ESMF architecture for the NUOPC operational centers and participating organizations. The CDT will develop the software and software interfaces necessary to modify ESMF for NUOPC applications. This NUOPC Layer will be distinct from the ESMF main distribution.

The CDT will receive guidance from the Content Standards Committee (CSC) on priorities and requirements on how to implement the NUOPC Layer.

The member of the NUOPC staff responsible for the Common Model Architecture will monitor the CDT. The manager of the CDT should be an ex officio member of the ESMF change review board. The CDT will be composed of coding specialists involved in the development and implementation of ESMF and the NUOPC Layer. The CDT should have input to the CSC and work closely with them in the NUOPC Layer implementation. At least one person from the CDT should be a CSC member. The CDT should be able to meet and advise with people from the operational centers and development laboratories as necessary.

Future Model Architecture Committee Charter

The NUOPC Future Model Architecture (FMA) Committee will define a common physical architecture that enables greater interoperability of a specific set of weather prediction components. The *physical architecture* is defined as the scientific scope of components – what processes they include – and the relationship of components to each other in the modeling system.

The member of the NUOPC staff responsible for the Common Model Architecture will also chair the FMA Committee. This committee will be composed of members of model development organizations, notably NCEP/EMC, NRLMRY, NASA GMAO, GFDL, ESRL, the tri-agency operational centers, NCAR ESMF & MMM, and perhaps other university groups. The FMA Committee will interact with the CSC to evolve standards suitable for future systems. The FMA Committee will interact with the Committee for Operational Processing Centers (COPC) on recommendations for computer hardware requirements and purchases.

The development of a future model architecture is expected to be an incremental and exploratory process, and will start with the coupling of major model components. The

FMA Committee will meet initially to determine how to proceed, and then quarterly thereafter, by teleconference and/or internet with periodic face-to-face meetings.

APPENDIX IV
COMMON MODEL ARCHITECTURE COMMITTEE TASKS

CSC – Content Standards Committee; FMA – Future Model Architecture Committee; MGMT – NUOPC management; OP CTR – operational centers

RECOMMENDATIONS	CSC	Core Team	FMA	MGMT	OP CTR
REC1.1.1 Adopt the NCAR ESMF Modeling Framework at the model-to-model level.					X
REC1.1.2 Establish a convention for data ownership in coupling interactions and adopt in application codes.	X	X			X
REC1.1.3 Create software templates for Gridded Components and drivers for use in NUOPC applications and adopt in application codes.	X	X			X
REC1.2.1 Decide conventions for templates for Component Couplers and adopt in application codes.	X	X			X
REC1.3.1 Establish conventions for use of Clocks and adopt in application codes.	X	X			X
REC1.4.1 Represent data at the Field level in inter-component interactions and adopt in application codes.					X
REC1.5.1 Decide on sufficient component documentation and implement in application codes.	X				X
REC1.5.2 Decide on sufficient component metadata, following the METAFOR CIM, and adopt in application codes.	X	X			X
REC1.5.3 Establish preliminary Field and FieldBundle metadata conventions and adopt in application codes.	X	X			X
REC1.6.1 Decide on convention for format of externally calculated interpolation weights, and utilize common remapping routines in application codes.	X	X			X
REC1.7.1 Design or adopt a shared software package that will read and write both netCDF4		X			X

and GRIB2 data formats and adopt in application codes.					
REC1.8.1 Design shared read/write component templates that use a FieldBundle write interface internally for large-scale asynchronous I/O and adopt in application codes.		X			X
REC1.9.1 Design shared diagnostics and post-processing tools and conventions and adopt in application codes.	X	X			X
REC1.10.1 Explore whether common configuration files are desired or feasible and if so implement them. Modify NUOPC applications to follow the conventions.	X	X			X
REC1.11.1 Design a unit test code for each NUOPC model or component.	X				X
REC1.12.1 Establish portability requirements for NUOPC codes and modify applications to satisfy requirements.	X				X
REC2.1.1 Define a common physical architecture for components with ESMF interfaces and implement in application codes.			X		X
REC2.2.1 Create a Physical Constants Module and implement in application codes.	X	X			X
REC3.1.1 Maintain ESMF, including resources for testing, documentation, development in response to bug reports and feature requests, repository, trackers, etc.		X			
REC3.2.1 Develop and update ESMF capabilities in response to application requirements.		X			
REC3.3.1 Monitor ESMF for performance and memory use and optimize the framework for new computing architectures. Impose no more than 5% penalty.		X			
REC3.4.1 After 2010 ESMF interfaces should be backwards compatible following the ESMF 5.0 public release for indefinite period of time. Determine a policy for exceptions.		X		X	
REC3.4.2 Monitor NUOPC participants'				X	X

compliance in migrating to new ESMF public releases within one year of release.					
REC3.4.3 Monitor NUOPC participants' compliance with ESMF release support policy.				X	X
REC3.5.1 Provide support for installation, design, and implementation questions concerning ESMF and NUOPC infrastructure.		X			
REC3.6.1 Decide on tutorials and higher-level documentation that describe how to use the framework for new users.		X			X
REC3.7.1 Provide longer-term, direct assistance for NUOPC participants in implementing ESMF in application codes.		X			X
REC3.8.1 Create a generic component compliance checker to see if component is ESMF compatible and confirm proper operation with application groups.		X			X
REC4.1.1 Adopt Subversion as the NUOPC configuration management software and implement in ESMF and applications.		X			X
REC4.1.3 Decide on way that all code components are accessible through a common interface such as a single web page or web interface and implement.	X	X		X	X
REC4.1.4 Decide on plan so that configuration management structure provides multiple levels for access for both reading and modifying component code and implement.	X			X	X
REC4.2.1 Decide how to offer NUOPC templates and codes via a distribution portal and implement.	X	X		X	X
REC4.3.1 Develop clear security standards and guidelines and adopt in application codes and ESMF.	X	X			X
REC5.1.1 Adopt a generalized vertical coordinate index so that new subroutines with a vertical index can be used with coordinate K=1 at top or bottom.					X
REC5.1.2 Avoid the use of common blocks, global data statements, and coding methods					X

that result in variables being passed between ESMF coupled components via shared memory.					
REC6.1.1 Extend the ESMF Core team so that it is responsible for building the NUOPC layer.		X		X	
REC6.1.2 Create a NUOPC Content Standards Committee with the responsibility for developing conventions for NUOPC codes.	X			X	
REC6.1.3 Establish a Future Model Architecture Committee to investigate/coordinate what physical processes future models contain and how they inter-relate			X	X	
REC6.1.4 Create a broad forum to encourage participation among NUOPC modeling groups.		X		X	