# Table of Contents

1 Scientific/Technical/Management  
   1.1 Objectives and Expected Significance  
      1.1.1 Objectives  
      1.1.2 Relevance to NASA Program Elements  
      1.1.3 Expected Significance  
   1.2 Technical Approach and Methodology  
      1.2.1 Testing  
   1.3 Perceived Impact to State of Knowledge  
   1.4 Work Plan  
      1.4.1 Key Milestones  
      1.4.2 Management Structure  
      1.4.3 Contributions of Principal Investigator and Key Personnel  
      1.4.4 Collaborators and Consultants  
      1.4.5 Risk Management  
      1.4.6 Management of Source Code and Information  
2 References and Citations  
3 Biographical Sketches  
4 Current and Pending Support  
   4.1 Current Awards  
   4.2 Pending Awards  
5 Budget Justification  
   5.1 University Of Colorado Budget Narrative  
   5.2 University of Colorado Budget Details  
   5.3 Georgia Institute of Technology Budget Narrative (Subcontract)  
   5.4 Georgia Institute of Technology Budget Details (Subcontract)  
   5.5 NASA GSFC Budget Narrative  
   5.6 NASA GSFC Budget Details
1 Scientific/Technical/Management

The work outlined in this proposal adds capabilities to and increases the robustness of the Cupid Integrated Development Environment (IDE) with a focus on adding new tooling to support development, execution, and runtime analysis of the Goddard Earth Observing System Model, Version 5 (GEOS-5). The work will result in a set of intelligent, integrated development tools that increase the overall usability of GEOS-5, simplify model execution and debugging, provide visualization of coupling control and data flows, and improve interoperability of GEOS-5 with other models based on the Earth System Modeling Framework. Cupid has been previously supported by CMAC funding (under ROSES 2011) and the proposed work builds on its existing capabilities, ensures robustness and backward compatibility with NASA’s Model E climate model, supported under ROSES 2011, and broadens its applicability with new support for GEOS-5. The work takes a critical step toward workflow consistency and code interoperability and opens the door to cross-training at the two centers.

IDEs are advanced software engineering tools that improve developer productivity and help to manage large codebases and complex software engineering processes involving a diverse set of tools. Cupid extends the capabilities of the popular Eclipse IDE, adding customized tooling for geoscience models based on the Earth System Modeling Framework (ESMF) and the National Unified Operational Prediction Capability (NUOPC) software layer. ESMF is a software framework for building and coupling Earth system models. NUOPC is an additional software layer, on top of ESMF, designed to increase interoperability of ESMF applications by defining a set of generic software components and technical rules for how to connect them. NUOPC is being implemented in the Community Earth System Model (CESM), Navy regional and global coupled models, the NOAA Environmental Modeling System (NEMS), and Model E at NASA GISS. Because NUOPC has been incorporated into these models, it becomes much easier to implement coupled models made up of components from these different systems and institutions.

Cupid’s existing tooling, built under ROSES 2011, includes a static source code analysis engine and code generation facility that simplifies creation and modification of ESMF and NUOPC model components. Cupid’s target user base includes model developers, especially those making architectural or infrastructure-level changes, such as introducing new model components or modifying the coupling configuration. Currently, model code can be edited and compiled from within the IDE and, through remote file synchronization, can be executed on HPC platforms such as NASA’s Discover. Additionally, Cupid includes a “wizard” interface for integrating the IDE with cloud platforms, such as Amazon’s Elastic Compute Cloud (EC2).

In conjunction with the IDE work described below, infrastructure-level improvements were made to better modularize Model E under previous CMAC funding, including the addition of ESMF and NUOPC interfaces to Model E’s ocean and atmospheric models. These changes allow Model E to take advantage of Cupid’s code analysis and verification functions and couple with other NUOPC-based models.

Here, we propose to extend Cupid’s existing tools, which are static in nature, with improved capabilities for executing, monitoring, and debugging ESMF applications. This includes the addition of control and data flow visualizations and improved compliance checking capabilities. The resulting tools will simplify the process of verifying correct
model coupling implementations and will provide a richer, more informative user interface for executing and debugging ESMF-based models. Additionally, we will optimize and improve Cupid’s existing code analysis and code generation engine, and improve its technical documentation and overall robustness. GEOS-5 will serve as the primary test application for the proposed new features. At the same time, to promote continuity, we will continue our active partnership with the Model E team, addressing practical issues that hinder use of the IDE for end-to-end workflows and continuing the Model E infrastructure improvements started under ROSES 2011.

In parallel with Cupid development, we will make improvements to GEOS-5 infrastructure to ensure its convergence with the latest developments of ESMF and the NUOPC interoperability layer. We propose to unify overlapping portions of the Modeling Analysis and Prediction Layer (MAPL) used in GEOS-5 with NUOPC. The resulting unified infrastructure will provide significant technical and scientific advantages to GEOS-5, including making the latest ESMF features available to the model (e.g., unstructured grid remapping), improving GEOS-5’s coupling interoperability with Model E, NCEP, GFDL, and Navy models, and reducing duplication of effort associated with maintaining multiple infrastructures.

Finally, to address the lack of availability of training materials about Earth system modeling infrastructure, we will team with Georgia Tech Professional Education (GTPE) to design and produce a set of on-line training modules about infrastructure for Earth system models and use of the Cupid IDE for model development. These materials will be made available for trainees and the general scientific community to use at a self-directed pace, and will complement traditional, face-to-face workshops.

1.1 Objectives and Expected Significance

1.1.1 Objectives

OBJ1. Address practical issues that limit Cupid’s usability in realistic, end-to-end development workflows by working through the deployment process with NASA partners.

OBJ2. Expand Cupid’s support for executing, monitoring, and debugging ESMF applications, including supporting the GEOS-5 development workflow within the IDE and adding coupled model control flow and data flow visualizations.

OBJ3. Unify overlapping parts of the MAPL and NUOPC infrastructure layers resulting in a GEOS-5 capable of coupling to NUOPC-based components and applications.

OBJ4. In collaboration with Georgia Tech Professional Education, design, produce, and publish a set of interactive, online training modules about ESMF modeling infrastructure and the Cupid IDE.

1.1.2 Relevance to NASA Program Elements

The objectives align with these specific efforts of interest appearing in the ROSES solicitation, listed below in italics:

ROSES 2014 A40 Section 2: To increase the scientific productivity, an Earth system science researcher’s workbench (or integrated development environment) is necessary
to integrate model software development environment, experiment execution, data analysis and visualization workflows.

ROSES 2014 A40 Section 2.2: The use of commercial and open source workbench, also known as integrated development environments (IDE), has greatly expanded in various software development communities. However, the components to support the use of Earth System Modeling Framework ... and other tools, such as compiler, debugger, restructuring tool, and source code version control systems used for developing parallel codes on a supercomputer are not commonly available. An IDE will become more and more important as the modeling community adopting modeling frameworks and collaborative architectures such as the National Earth System Prediction Capability... Funding will be provided to develop an open source IDE that will be available especially to the GEOS-5.

The work proposed directly addresses the development of an IDE that integrates the model software development environment with the experiment execution workflow. Eclipse is one of the most mature IDE platforms currently available and enjoys very active user and developer communities. The Parallel Tools Platform⁶ and Photran⁶ plugins offer specialized tooling for developing parallel codes written in Fortran and could be of benefit to the geoscience modeling community. Up until this point, we have, by and large, not experienced adoption of these tools at modeling centers within NASA or elsewhere. While this lack of adoption may be attributed partially to cultural factors, we suspect the nuanced nature of geoscience model development is also a key factor, especially the sheer size and complexity of geoscience codebases, the large number of configurations that must be supported, and the specialized software processes required for model verification and validation.

Cupid, its existing capabilities and the improvements proposed here, builds on the general-purpose IDE functions previously mentioned and offers domain-specific capabilities of tangible value to model developers: automated code generation, built-in compliance checking and verification, model control flows, coupling field data flows, model-specific documentation, and the like. Because Cupid brings these domain-specific functions and builds on Eclipse’s existing, well-supported infrastructure, it represents the most sustainable path forward for delivering an IDE that could realistically benefit Earth system modelers at NASA.

ROSES 2014 A40 Section 2: This program is also concerned about the interdisciplinary workforce development, especially at the interface between Earth, computing and computational sciences, and software engineering. Funding will be provided to develop an annual interactive workshop and internship in specialized computational and software engineering techniques for Earth system modeling and data analysis to supplement the higher education programs at research universities.

Although we do not propose an annual workshop here, the proposed work complements a comprehensive training workshop for environmental scientists in several ways. First, the Cupid IDE reduces the technical expertise required to work with Earth system model codes, and provides a more intuitive approach to running models compared with command line-based tools, especially for graduate students or trainees that are new to HPC environments. Cupid, therefore, can be used in workshop settings as a simplified training environment, allowing students to perform basic modeling functions without comprehensive knowledge about high performance computing (HPC) and computer science. Cupid’s basis on the popular Eclipse platform introduces next
generation geoscientists to modern software engineering tools and practices, especially use of source control, smart code editors, and advanced debugging and testing tools.

Cupid also features a connection with Amazon’s Elastic Compute Cloud (EC2) which allows users to spin up machine images that are pre-configured with Earth system modeling software and its dependencies. This offloads resource-intensive system configuration and administration processes to the cloud and provides trainees with low-risk sandboxes for learning how to work with a model codebase.

Finally, the online training materials that we will produce complement a facilitated, face-to-face workshop with a set of self-directed, interactive modules that allow students to work at their own pace, to easily review previous lessons, and to master material before moving on to the next lesson. These materials could be used in conjunction with a workshop or as an additional educational resource for attendees either before or after the workshop.

1.1.3 Expected Significance

- We expect that IDEs will play an increasingly important role in the development of Earth science modeling applications, eventually supplanting purely command-line driven approaches. We also expect HPC platforms to continue to move toward heterogeneous architectures such that software engineering tools will become more critical in managing code complexity and multi-step compilation processes. With these in mind, the proposed usability improvements to Cupid will help to break down technical barriers that currently impede adoption of IDEs at NASA and other modeling centers. The domain-specific capabilities will attract users by providing tools that “speak their language” and the improved technical documentation and training materials will help to bridge existing knowledge gaps.

- We expect that the proposed dynamic analyses, including model control and data flow visualizations, will provide significant benefit to developers using ESMF by providing novel, dynamic views of multi-component, parallel coupled models that aid in debugging. More generally, this will help Earth science modelers better understand the benefits of an IDE equipped with dynamic analysis tooling.

- We expect to build on the relationships already established with key leadership in the Eclipse community, including project leads and developers from the Parallel Tools Platform and Photran, the Fortran language tool for Eclipse. Under ROSES 2011 we helped define and prioritize improvements to Eclipse infrastructure that are critical for the IDE’s success is the Earth sciences. These relationships will be vital going forward: they ensure the ESM community has a voice in defining new requirements for Eclipse and prioritizing development activities.

- We will offer an alternative development workflow for GEOS-5, based on Eclipse, from code acquisition to model execution. We expect that use of an IDE with GEOS-5 will both simplify some aspects of the workflow, and also stretch Eclipse due to the size and specialized workflow requirements of GEOS-5.

- We expect that the integration of the MAPL and NUOPC infrastructure layers will reduce the amount of time required to couple GEOS-5 with NUOPC components, including NUOPC components of Model E. Additionally, by unifying overlapping parts of these infrastructures, we reduce duplication of code and maintenance
effort, and ensure that GEOS-5 can take advantage of the latest ESMF features and its robust testing processes.

- We expect that the interactive, web-based training modules will offer an appealing, experiential learning opportunity not currently available for training on geoscience modeling infrastructure.

1.2 Technical Approach and Methodology

The technical approach will follow the established Cupid development process, including use of its source repository, coding standards, test harness, and release procedure. This technical approach is based on ESMF’s development process that has evolved over its lifetime and is described in the ESMF Developer’s Guide. We expect to continue to engage with the broader Eclipse community, especially technical leads of the Parallel Tools Platform and Photran plug-ins. We will remain active on the Eclipse forums and will utilize already established issue trackers for filing bugs and feature requests when necessary. During our interactions, we will represent the needs of the broader geoscience modeling community.

OBJ1. Address practical issues that limit Cupid’s usability in realistic, end-to-end development workflows

The first objective is to reduce technical barriers to adoption of the IDE by ensuring that the software possesses the capabilities needed for practical use and has been deployed and thoroughly tested in realistic development contexts at NASA centers. We will work in tight collaboration with technical leads and developers of both Model E and GEOS-5 to ensure that features developed in relative isolation are usable in an end-to-end workflow on NASA HPC platforms. The scope of the development workflow includes: checking out model code from source control, synchronizing files between HPC clusters and the user’s local machine, model configuration for a particular experiment (either via scripts or Make), compilation, model execution, either locally or via submission to the batch queuing system, and verification of successful execution via log files or standard output.

Practical usability issues that we have already identified appear below. Additionally, we will build on infrastructure improvements made to Model E under ROSES 2011, also outlined below. We will address any unforeseen issues while working through Cupid’s deployment at NASA GSFC. The resulting improvements will raise the Technology Readiness Level of Cupid by moving from representative model codes on modest computing resources, towards validation in a relevant environment.

The first version of Cupid, 0.1 beta, has been released and is available for download on the Cupid web site. Installation instructions and a tutorial are also available. The release includes (1) a code analysis engine capable of reverse engineering a model’s codebase, (2) a code generation function for generating code templates required by ESMF/NUOPC, and (3) a “wizard” interface that connects the IDE automatically to a cloud computing platform with pre-configured machine images used for training on model development. We now briefly describe the status of Cupid’s existing capabilities and outline the proposed usability and optimization activities:

- NUOPC infrastructure has been added to Model E including a new top-level driver as well as NUOPC versions of the atmospheric and ocean components. Currently, NUOPC is being used to drive an active atmosphere with mixed-
layer ocean coupling configuration. This new infrastructure is not fully integrated with Model E—a limited set of fields are exchanged. Going forward, we propose to extend this work by fully integrating NUOPC into Model E, resulting in a well-modularized, NUOPC-compliant codebase.

- Cupid’s reverse engineering function parses a model’s codebase and produces a high-level view of the model’s architecture directly beside the source code in the IDE. This high-level view helps a developer understand and navigate a model’s code, highlights any compliance issues found (e.g., missing code expected by the framework), and provides contextual framework documentation from the NUOPC reference manual. The reverse engineering function is currently un-optimized, requiring several minutes execution time on larger codebases (hundreds of files). In the interim, a compromise was introduced to work at a file-at-a-time basis, instead of parsing an entire codebase. This negatively impacts usability. Therefore, we propose to optimize this function by introducing heuristics and a fast code index. This is expected to provide significant speedup, such that reverse engineering happens nearly instantaneously as code is written.

- The IDE has been tested with modest computing resources, such as developer laptops and servers. Progress has also been made in connecting the IDE with NASA Discover. This required the addition of new SSH proxy capabilities to deal with Discover’s multi-factor authentication. The next steps we propose are to integrate with the batch queuing system and to expand to additional HPC platforms, including the Pleiades cluster used by GEOS-5.

- Cupid’s code generation function generates NUOPC-compliant code templates in situ based on the current state of existing code. The generated templates are highlighted in the code editor and can be further customized by the developer. The primary maintenance work required is to ensure that Cupid remains up to date with the latest ESMF/NUOPC API changes. As new versions of ESMF are released, we will ensure Cupid remains current so that generated code will compile with the most current ESMF library.

- Cupid features a “wizard” interface that automatically connects the IDE to a virtual machine instance running on Amazon’s EC2 cloud platform. Cloud access is currently available only internally because it requires access to the ESMF EC2 account. When cloud access is required for using Cupid in a training course (see objective four), then we will provide a mechanism for opening up this access to Cupid users.

- Cupid’s features are documented in a feature overview and tutorial document. We propose to introduce a more comprehensive user’s manual, including detailed descriptions of all features, as well as model-specific documentation for performing development workflows with Model E and GEOS-5.

**OBJ2. Expand Cupid’s support for executing, monitoring, and debugging ESMF applications**

Cupid’s existing functions are mostly based on static analysis techniques—i.e., analyzing code structures without execution. The static analyses performed by Cupid are able to detect many potential or definite issues that lead to model component interoperability issues or failed execution altogether. An advantage of this technique is
that problems can be found early on, as code is being written, and corrected before expending compute resources attempting a run. However, static analyses are limited because it is usually difficult or impossible to predict all possible execution paths. With respect to the geoscience models that Cupid supports, aspects limiting static analysis include extensive use of conditional compilation via the C preprocessor and the use of complex configuration scripts prior to model execution. Because of this, Cupid’s reverse engineering function can suffer from false positives (incorrectly determining an ESMF application is “correct,” i.e., compliant with NUOPC technical rules) and false negatives (incorrectly determining that a correct application is faulty).

Here, we propose to add dynamic analysis capabilities to Cupid that will complement the existing static analysis tooling. This will result in better runtime support for model developers, including detailed visualizations of the independent control flow paths of model components in a coupled application and field-level data flow analyses showing data movements relevant to coupling. Understanding the control sequence among the components and the flow of data through the system is essential for verifying correct construction of a multi-component geoscience model. Many factors, including code complexity, the increasing number of coupled components, and the requirement to support a large number of model configurations make it difficult to understand a coupled model’s behavior simply by inspecting its code. This complicates and slows down model debugging and delays execution of scientifically meaningful runs.

The new runtime features will be based on tracing. Tracing is a technique used to understand the behavior of a running system, and software that performs tracing is called a tracer. Conceptually, a tracer is analogous to a tape recorder that monitors and records events about a running system. The output is a trace file that is analyzed to produce statistics about the system’s dynamic behavior. Most existing tools that provide tracing capabilities for massively parallel codes, such as Scalasca, Score-P, and TAU, are general purpose tools designed especially for performance tuning, such as identifying parts of the application that are poorly load balanced. As such, the event stream is usually low-level, capturing individual MPI send and receive events. While this level of fine-grained detail is important for performance optimization, abstraction layers provided by ESMF are lost during the trace. This complicates the ability to produce application level analysis in terms of ESMF structures: Gridded and Coupler components, Fields, Grids, and States. Additionally, off-the-shelf tracing tools have steep learning curves and usually require extra compilation steps to instrument codes to record event traces.

Therefore, in lieu of off-the-shelf packages, we will build on existing ESMF functionality including the ESMF logging facility and the Compliance Checker. The logging facility outputs either a single log file or multiple log files (one per process) including both framework-level and user-defined events. The Compliance Checker is an existing tool that can be enabled via an environment variable. When enabled, this tool augments the standard log output with additional details about compliance with NUOPC interoperability rules. These tools could be used to produce an application-level event stream of an appropriate granularity for coupled model control and data flow analyses.

The Compliance Checker will be modified, as necessary, to record events related to the execution of ESMF methods (init, run, finalize) as well as the contents of ESMF State objects (coupling fields) as they are transferred between components. This tracing
The technique is appealing due to its simplicity, and because it does not require existing ESMF users to install a third-party tracing tool. Three user interfaces will be added to Cupid to display analyses derived from the application-level event stream: an ESMF log viewer, a control flow viewer, and a data flow viewer.

The ESMF log viewer will parse ESMF log files and provide an event trace in table format inside the IDE. The viewer will be populated either by manually selecting a previously generated log file, or by configuring the viewer to automatically populate immediately after termination of an ESMF application. A configuration page will allow the user to highlight warnings and errors that appear in the log and filter the log by event type, process number, and/or ESMF component.

![Figure 1 - Control Flow](image)

A control flow view will be added to the IDE that shows the dynamic behavior of each component in a coupled model. Figure 1 depicts a screen mockup of a control flow analysis. The hierarchical nature of the coupled application can be seen in the column labelled “Component.” The top-level component GEOSgcm has three direct child components: GEOSagcm, GEOSogcm, and Atm2OcnMed. Parallel components that execute on multiple processes can be collapsed into a single line with the number of processes listed in the “PET Count” column. In Figure 1, the top-level component, GEOSgcm, executes on a total of 1024 processes. Of those 1024 processes, the GEOSagcm child component utilizes 768 for its execution while the GEOSogcm uses 256. The colored Gantt chart on the right indicates when ESMF phases are executed and shows whether child components are executing concurrently or sequentially. Model-specific metadata is shown for each execution phase, such as the current model time and contents of the model’s import and export States.

A data flow view will be added to Cupid that shows the model components participating in a coupled application annotated with directed arrows showing data flow of coupling fields. This analysis will assume a fixed exchange pattern that repeats every coupling interval, which is typical in geoscience models. Therefore, the data flow visualization will only require executing the configured model for one full coupling cycle. The data flow viewer will allow the user to drill down to a pair of models and examine field-level metadata collected by ESMF.
Where possible, we will leverage existing Eclipse plug-ins to build the dynamic analysis back end and visualization components. One promising plug-in, Trace Compass, offers an extensible framework for processing and displaying application event logs and traces. It supports several popular trace formats natively, and provides the Trace Monitoring Framework (TMF), which facilitates integration of user-defined trace data into the Eclipse IDE. We will determine whether this tool can be extended to process ESMF log files, either in their current form, or after pre-processing.

**OBJ3. Unify overlapping parts of the MAPL and NUOPC infrastructure layers**

GEOS-5 is built on with a software infrastructure layer called the Modeling, Analysis and Prediction Layer (MAPL), created at NASA GMAO, that simplifies and constrains the creation of ESMF components. MAPL includes both generic component constructs as well as model-specific code. Work has already begun to find a path for interoperability between MAPL and NUOPC, i.e., using MAPL components in NUOPC systems and using NUOPC components in MAPL systems. In the former case, the GEOS-5 atmosphere could be converted from a MAPL component into a NUOPC component in order to participate in a multi-model, interactive ensemble. In the latter case, the WaveWatch III model, which is being converted into a NUOPC component, could be coupled into GEOS-5. An increasing number of NUOPC-compliant components are being made available within an emerging Earth System Prediction Suite (ESPS). By providing a convergent path for MAPL and NUOPC, GEOS-5 will be able to interact with components in the ESPS.

An initial analysis performed in January 2013 showed that MAPL and NUOPC are fundamentally similar in structure and capabilities and are therefore highly compatible. Especially relevant is the fact that both NUOPC and MAPL wrap native ESMF data structures (ESMF_GridComp and ESMF_CplComp) and do not change their fundamental behaviors or interfaces.

The analysis also indicated some differences that must be reconciled. GEOS-5 uses ESMF components in a more fine-grained manner than NUOPC applications. Currently, NUOPC conventions are targeted at the highest level components in a coupled system, e.g., atmosphere, ocean, ice, and wave components. In GEOS-5, ESMF components extend to a large number of deeply nested components, including processes such as atmospheric radiation, chemistry, and microphysics. While both MAPL and NUOPC can support hierarchies of components, they are each specialized for their target applications. For example, NUOPC Drivers are separate generic components capable of driving a set of child components with little or no specialization (additional code). Also, within NUOPC, custom coupling code is included in Mediator components. MAPL driving code is interspersed with coupling code and both are specific to a particular set of child components. Part of the API reconciliation, therefore, will be to ensure that NUOPC generic Drivers are capable of handling the specialized driving code found at all layers of GEOS-5. It may be necessary to introduce new kinds of NUOPC generic Drivers, or to introduce more specialization points for fine-tuning the behavior of a generic Driver.

Another difference that must be reconciled is the specialized way in which MAPL implements matching of import and export fields across a deeply nested hierarchy of components. In MAPL, export fields are swept up the hierarchy, propagating from leaf
components upwards such that a parent component aggregates export fields from its children. Unsatisfied imports also propagate up the hierarchy to determine if they can be satisfied by a higher level component, or ultimately, with data in an external file. For matching coupling fields, NUOPC provides a field dictionary containing standard names and a field brokering process for automatically matching import and export fields. However, this procedure works only on sibling components at the same level in the hierarchy. This specialized behavior of MAPL will either need to be introduced as a NUOPC-supported capability or abstracted into a specialized layer specific to GEOS-5. The former is slightly preferable as other modeling systems would likely benefit from a child-to-parent field propagation capability.

Before API reconciliation, preliminary work will be performed, including upgrading GEOS-5 to use version 7 of ESMF as well as a more detailed technical analysis to determine which capabilities will be merged with NUOPC and which will remain GEOS-5 specific.

**OBJ4. Produce interactive, online training modules on Earth system modeling infrastructure**

We will make available a set of interactive, online training modules aimed at bridging the knowledge gap between standard graduate training in environmental science and the realities of working with the complex cyberinfrastructure used for modern Earth system modeling. The training materials we will develop focus specifically on ESMF modeling infrastructure, the NUOPC interoperability layer, and using the Eclipse IDE with Cupid tooling. The result will be a learning experience that is much more engaging than existing educational resources such as tutorials, sample code, and static demo applications.

We will team with Georgia Tech Professional Education (GTPE), a leading organization in designing and producing instructional material for web-based delivery on Learning Management Systems such as Coursera\(^\text{12}\), Udacity\(^\text{13}\), and Moodle\(^\text{14}\). GTPE has prior experience developing educational material for NASA through the Electronic Professional Development Network and has created the NASA Virtual University\(^\text{15}\), which hosts both self-directed and facilitated courses for educators. GTPE will provide instructional design support for a series of five hands-on training modules about Earth system modeling infrastructure. The modules we design will rely on Cupid as a companion training tool. Participants will install Cupid locally and perform programming assignments in the IDE during each lesson. The course design will take advantage of Cupid’s ability to spin-up pre-configured virtual machine instances on Amazon’s cloud computing platform. These virtual machine instances will allow trainees to “get their feet wet” working with ESMF-based codes in a sandbox environment. For trainees with adequate credentials, recent improvements to Eclipse’s remote services plug-in also allows Cupid to connect with HPC resources such as NASA’s Discover.

We will use GTPE’s current course design and development process and the resulting modules will be hosted publicly on GTPE’s Helix\(^\text{16}\) Learning Management System. The following table outlines the planned training topics and their learning objectives.

<table>
<thead>
<tr>
<th>Module Topic</th>
<th>Learning Objectives</th>
</tr>
</thead>
</table>

Use or disclosure of information contained on this sheet is subject to the restriction on the cover page of this proposal.
| Introduction to the Earth System Modeling Framework | • Understand the purpose of ESMF and its basic capabilities  
• Understand the architecture of ESMF applications |
| Working with an Integrated Development Environment | • Learn what an IDE is and what tools it provides  
• Learn how to navigate the Eclipse IDE and the Cupid tooling  
• Learn to acquire code from source control, compile and execute it using Eclipse |
| Writing an ESMF Application | • Learn how to build a basic ESMF application |
| Adapting an Existing Model to ESMF | • Understand the steps involved in adapting an existing application to use ESMF |
| Introduction to the National Unified Operational Prediction Capability (NUOPC) Layer | • Learn the basic architectural components of NUOPC: Models, Drivers, Mediators, and Connectors  
• Learn how to construct a NUOPC application from sets of components |

1.2.1 Testing

The proposed work includes multiple kinds of software that will each require different testing strategies as outlined below.

An automated test suite is in place for Cupid’s main features. The test suite performs reverse engineering and code generation system tests and exercises the GUI components using JUnit\(^\text{17}\) and the SWTBot\(^\text{18}\) testing apparatus. Consistent with ESMF development practices, we will employ a “test first” strategy in which the existing test suite is extended with new tests before implementing new functionality. The test suite is run during development and on all supported platforms (Linux, Mac, and Windows) before all releases. A comprehensive release checklist\(^\text{19}\) has been formulated outlining the steps required before a release.

Because of the highly interactive nature of IDEs, it is difficult to automate tests that adequately represent realistic development workflows. Therefore, we will put several measures in place to ensure that the developed interfaces are intuitive and match user expectation: We will communicate frequently with the NASA Model E and GEOS-5 developers collaborating on this project, including scheduling demos with screen sharing and providing opportunities for feedback. During these sessions, some time will be dedicated to allowing external users to “sit in the driver’s seat,” describing their interactions with the tool, and noting places where expectations were not met. Detailed technical documentation, including graphical tutorials, will be provided that describe the intended function of Cupid. Finally, we will clearly indicate, in release notes and on the Cupid web site, the current status of Cupid’s features, carefully noting known bugs and uses of the IDE that do not yet work properly or might be confusing. The process will be one of setting user expectations, training, and gathering feedback to improve the tool, where possible.

Testing the unification of MAPL and NUOPC in GEOS-5 will require a different kind of testing. The ESMF team has significant experience making infrastructure-level
changes to model codes and has an established process that ensures, in almost all cases, bit-for-bit reproducibility of previously validated model output. Basic validations of the unified MAPL/NUOPC API can be performed outside of GEOS-5 using simple prototype applications, similar to the existing NUOPC prototypes\(^2\). A minimal set of tests will be determined and agreed upon by GEOS-5 technical leads before making modifications to GEOS-5, and that work will take place on a separate branch. Deliveries of a set of software changes to MAPL and GEOS-5 will be accompanied by a report that includes all relevant software versions, computing platforms, a list of tests that were run and their results, and any information required to acquire the modified software and to reproduce the tests. The GEOS-5 team will be encouraged to run their standard acceptance tests against the modified software.

1.3 **Perceived Impact to State of Knowledge**

The proposed work is a practical exploration into how advanced development environments can be used by model developers to enhance productivity, manage an array of diverse tools, and ultimately take better advantage of cyberinfrastructure resources used for environmental prediction. The knowledge contribution of this project is a better understanding of the advantages and disadvantages of the IDE approach to model development as well as new technical expertise for what it takes to build domain-specific development tools that provide true value to NASA modelers.

To our knowledge, no modeling center, within NASA or elsewhere, is using IDE-based approaches comprehensively—most rely on customized build systems, text-based configuration files, shell scripts, and an array of disparate post-processing and analysis tools. Admittedly, most IDEs are not designed for scientific software development, although steps are now being taken to retrofit them for the needs of scientific software engineers. And, while existing development workflows are functional and pragmatic, complexity continues to rise at an alarming rate, both on the software and hardware sides. IDEs encourage a more holistic approach to software development and we hypothesize that IDEs will play an increasingly important role in the geosciences to help simplify workflows and provide a more appealing working environment.

1.4 **Work Plan**

The four objectives outlined above require one full time software engineer to support Cupid development at University of Colorado, a half-time post-doctoral researcher at NASA GISS to support collaborations involving Model E and a 75% time senior software engineer at NASA GSFC to support collaborations involving GEOS-5. Some additional support will be provided by ESMF lead developers, at no cost to this proposal, for technical guidance in the design and implementation of the unified MAPL/NUOPC layer.

The nature of the proposed work, and the lack of critical inter-dependencies, allows several tasks to proceed concurrently up until the point of integration:

- The proposed improvements to usability of Cupid’s existing tools are largely independent of the other tasks. The key collaboration will be with the post-doctoral researcher at NASA GISS who serve as a local expert for executing Model E workflows within the IDE. We will establish a frequent release process
and feedback loop such that modifications to Cupid are tested frequently against Model E.

- The dynamic analysis tooling is dependent only on obtaining ESMF log traces and the traces are likely useable in their current form, or with minimal pre-processing, at least for initial development of the new tools. This work can therefore start immediately at the beginning of the period of performance.
- The MAPL/NUOPC integration is largely independent of the Cupid development work and the initial analysis for this activity has already been completed under separate funding. Due to the deeply embedded nature of MAPL within GEOS5, changes to this infrastructure will require substantial collaboration with Co-I da Silva and the senior software engineer. We will initiate this process by identifying test cases and an incremental implementation strategy. We expect a majority of the implementation will be performed on the ESMF side with additional support from the senior software engineer.
- Production of the training modules will proceed under the processes already in place at GTPE: planning (creating course timeline), course design (determining technology needs, storyboarding, scripting), development of instructional and multimedia content (including editing), quality verification, and delivery. We will use the established ESMF development process, in which design and implementation reviews are conducted using distributed development approaches (conference calls, desktop sharing software, mailing lists, etc.). We will work with Model E and GEOS-5 users to evolve Cupid to the point at which it can be distributed and supported more broadly. The eventual goal is to demonstrate value and petition for Cupid to be distributed through the ESMF core development team and maintained through its multi-agency sponsorship.

### 1.4.1 Key Milestones

The anticipated start date for the proposed work is FY 2014-4Q. Cupid 0.1 beta has been released and additional releases are anticipated before the start of this proposal’s period of performance. We will assume the first release of Cupid under this proposal is version 0.5.

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY 2015-1Q</td>
<td><em>Release 0.5</em>: comprehensive user’s guide, Model E specific documentation, and updates according to NUOPC API changes</td>
</tr>
<tr>
<td>(October 2015)</td>
<td><em>Design document, implementation plan, and tests identified for MAPL/NUOPC unification</em></td>
</tr>
<tr>
<td></td>
<td><em>Initial course planning meeting with GTPE</em></td>
</tr>
<tr>
<td>FY 2015-2Q</td>
<td><em>ESMF log and Compliance Checker modifications complete</em></td>
</tr>
<tr>
<td>(January 2016)</td>
<td><em>MAPL/GEOS-5 upgraded to ESMF v7</em></td>
</tr>
<tr>
<td></td>
<td><em>Demonstrate basic IDE workflow with GEOS-5, including code import and compilation</em></td>
</tr>
</tbody>
</table>
| FY 2015-3Q (April 2016) | • Release 0.6: ESMF log viewer  
• Proof-of-concept unification of MAPL/NUOPC infrastructures including merge of MAPL component and NUOPC component concepts |
| FY 2015-4Q (July 2016) | • Release 0.7: optimizations for large codebases  
• Content finalized for training courses |
| FY 2016-1Q (October 2016) | • Unified MAPL/NUOPC layer integrated into GEOS-5  
• Media produced for training courses |
| FY 2016-2Q (January 2017) | • Release 0.8: data flow and control flow views  
• Post-production of training courses complete and materials posted online and accessible |
| FY 2016-3Q (April 2017) | • Demonstrate modified GEOS-5 coupled with external component through NUOPC  
• End-to-end development workflow demonstrated with GEOS-5 and Model E |

1.4.2 Management Structure

Dr. Ralph Dunlap of the University of Colorado is the PI of the proposed investigation. He is responsible for the execution of the proposed activity, the quality of products delivered, and the proper use of all awarded funds. He is also responsible for all technical, management, and budget issues and is the final authority for this project. The Co-Is report to and take direction from the PI and will provide the information needed to ensure that he can effectively manage the project.

As a member of the ESMF team, Dr. Dunlap will operate the project with oversight from Cecelia DeLuca, head of ESMF, and in accord with the overall strategic direction outlined in the 2010-2015 ESMF Project Plan.

1.4.3 Contributions of Principal Investigator and Key Personnel

PI Ralph Dunlap is a research lead within the ESMF team and is currently the lead designer and developer of Cupid. Dr. Dunlap will transition from his role as the primary Cupid developer into a managerial role while retaining some supporting development responsibilities. Dr. Dunlap will provide significant technical guidance to transition a new developer into the project, including guidance on understanding the architecture of Cupid, setting up the development environment, and running the test suite. In addition to directing the project, Dr. Dunlap will serve as principal technical architect of Cupid and will provide technical support as needed, including, but not limited to, writing documentation, maintaining and improving the codebase and ensuring its overall quality. Dr. Dunlap will present the software at strategic meetings, scientific conferences, and workshops to raise awareness and promote its use. Dr. Dunlap will also work with ESMF leads to develop content for the training courses.

1.4.4 Collaborators and Consultants

Dr. Arlindo da Silva is the NASA Goddard Institutional PI of the proposed investigation. Dr. da Silva was one of the PIs of the original ESMF proposal and has served in the ESMF Executive Board since its inception in 2000. Dr. da Silva currently leads the aerosol modeling and data assimilation efforts at NASA’s Global Modeling and Assimilation Office (GMAO), and manages its Software Infrastructure Team. He will
dedicate 5% of his time to this project and will be assisted by a part-time senior software enginee. Dr. da Silva's activities will be leveraged using GMAO Core funds, at no cost to this proposal. The GSFC Institutional PI is responsible for the execution of the NASA/GSFC portion proposed activity, the quality of products delivered, and the proper use of all funds awarded to GSFC. In particular, the GSFC team will provide technical support and testing of the GEOS-5 integration into Cupid, as well as assist in the MAPL/NUOPC unification activities.

Co-I Dr. Gavin Schmidt is the director of the Goddard Institute for Space Studies and one of lead developers of Model E. He will serve as a point of contact for Model E and will help to coordinate and provide technical guidance required for running Model E with Cupid.

Collaborator Dr. Thomas Clune is Chief of the Software Systems Support Office (SSSO). He and his staﬀ have contributed to software engineering and development of Model E, and will help to coordinate the Cupid work with continued development eﬀorts within NASA. Dr. Clune will help to inform how the Cupid development group can integrate and interact with model testing and will provide technical support in the continued infrastructure changes to Model E.

Consultant Dr. Leo Mark is associate dean for academic programs and student aﬀairs at Georgia Tech Professional Education and has oversight of all its academic offerings. Dr. Mark will serve as point of contact at GTPE and will coordinate the collaboration between the ESMF team and the instructional design staﬀ at GTPE.

1.4.5 Risk Management

The major risks associated with this project are that Cupid fails to provide real value to developers at NASA GSFC and that the unied MAPL/NUOPC layer is not integrated into GEOS-5.

Risk: Cupid fails to provide real value to developers. This could occur for multiple reasons. Practical usability issues within the IDE may be too many and too complex to resolve, resulting in a tool that works only in very limited contexts. Another possibility is that Cupid tooling does not address the real or perceived requirements of model developers. To mitigate, we will work with Model E and GEOS-5 in close collaboration, identifying as early as possible critical technical barriers to adoption, prioritizing those over more exploratory tasks.

Risk: The unied MAPL/NUOPC layer is not integrated into GEOS-5. This could occur for technical or cultural reasons. A key mitigating factor will be ensuring that the proposed modifications offer clear economic advantage and align well with scientific goals of the model. The modified infrastructure will be more likely to be adopted if it facilitates coupling with external components of interest to GEOS-5 scientists.

1.4.6 Management of Source Code and Information

Cupid is distributed under the MIT open source license and maintains a development repository and issue tracker that is publicly viewable. The Cupid team is committed to providing as much information about the project as feasible in an open, accessible way. All project documentation, and links to the software, are available on the Cupid web site: https://www.earthsystemcog.org/projects/cupid/