Data Registration, Match, and Model Component Coupling

Shujia Zhou*

NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA
shujia.zhou@nasa.gov

Carlos Cruz*

NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA
carlos.a.cruz@nasa.gov

Abstract

A coupling toolkit has been developed to reduce the complexity of model component coupling, in particular among hierarchical model components. The toolkit provides the services of data registration, data matching, data filtering, and model component coupling. In addition, it can generate diagrams to reveal the "producer"-to-"consumer" relations among the components. We have tested this toolkit with the operational NASA Goddard Earth Observing System Model, Version 5 (GEOS-5), which is built on the Earth System Modeling Framework (ESMF) and consists of multi-level Earth system components.

Categories and Subject Descriptors D.2.3 [**Software**]: Software engineering – coding tools and techniques.

General Terms Algorithms, Design.

Keywords Framework; component; coupling; data registration; data match; code generator

1. Introduction

The component technology is being used to couple model components from different organizations [1, 2], and decompose a complex application, such as a combustion model [3, 4].

One component-based framework is the Earth System Modeling Framework (ESMF), which attempts to standardize Earth system models and their couplings [5, 6, 7]. There are two kinds of ESMF components. One is the Gridded Component for a model, and the other is the Coupler Component for coupling two model components. They have no differences except the name at this time. ESMF is implemented with Fortran 95 (F95), C, and C++. It provides its functionality with a Fortran interface and with a C interface to some extent. In this paper, we use a Fortran interface when we discuss ESMF.

From the perspective of component coupling, an ESMF Gridded Component, implemented with an F95 module, has the following features:

- An ESMF component has three standard interfaces (F95 subroutines), ESMF_GridCompInitialize(…), ESMF_GridCompRun(…), and ESMF_GridCompFinalize(…).
- User's versions of these three routines are provided through ESMF_SetEntryPoint(…) and linked to these three standard interfaces with ESMF_SetService(…) (see Fig. 1).
- An ESMF component has only one import and one export state of ESMF_State, an F95 user defined data type (see Fig. 1). An ESMF import or export state serves as a container for all the import or export variables, their grid information, and other meta data.
- An ESMF coupler is used to exchange data between two sibling components (Fig. 2).
- An ESMF_State can be nested to support multiple model component coupling (see Fig. 3).
- An ESMF component can exchange data with multiple components through multiple couplers (see Fig. 3).

With ESMF, there are essentially three operations for model component coupling:

- Match the name of an export variable of a "producer" component with the name of an import variable of a "consumer" component.
- Transport the matched export variable to the "consumer" component.
- Transform the matched export variable as required by the "consumer" component, such as regridding (e.g., convert data from a coarse grid to a fine grid).

To couple two model components, such as atmosphere and ocean, an ESMF coupler is required to handle the data

*Employee of Northrop Grumman Corporation

Copyright 2007 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
HPC-GECO/CompFrame'07 October 21–22, 2007, Montréal, Québec, Canada.
Copyright © 2007 ACM 978-1-59593-867-1/07/0010…$5.00.

transfer for an export state (e.g., expAtm) to an import state (e.g., impOcn). Note that this coupler is located at the same level as the two model components to be coupled and these two model components are siblings, that is, they have the same parent (see Fig. 2). For the case where the component with export variables is in the same level as the component with import variables, it is straightforward for a coupler to perform data transfer and transformations such as regridding. However, it could be very challenging when coupling is among hierarchical components. For example, if u and v (longitudinal and latitudinal wind speeds) of a dynamics model component are exported to an ocean component, these two variables have to be packed into expDyn, the export state of the dynamics model component (dyn). expDyn is then passed to expAtm through expSdyn (the export state of the sdyn model component) if the feature of nested ESMF states is deployed for data transportation (see Fig. 4). Considering the fact that there can be tens of hierarchical model components and the export state of each component can contain tens of variables, it is clear that implementing this kind of coupling manually can be quite challenging. Therefore, a tool to reduce coupling complexity is highly desired.

2. Design

Since ESMF provides a standard way of coupling model components for Earth system models, we outlined a generic coupler for model components that use a standard naming convention such as the Climate and Forecast (CF) convention to name their export and import variables [8]. That design has been implemented and successfully used in coupling an atmospheric model, the Weather Research and Forecasting (WRF) model [9], with an ocean model [10]. However, that version needs to be augmented to deal with hierarchical components as found in the operational NASA Goddard Earth Observing System Model, Version 5 (GEOS-5). GEOS-5 [11] is built on ESMF and consists of multi-level Earth system components.

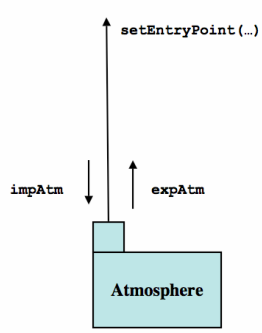

Figure 1. Schematic of an ESMF component, atmosphere model component. Its import and export states are impAtm and expAtm, respectively. Its external interface is setEntryPoint(…).

In this paper, we present the augmented design with one implementation. Essentially, there are four parts: data registration, data matching, data filtering, and a code generator.

2.1 Data registration

For a component to interact with other components, it needs to provide the following essential information:

- component name
- import variable names
- export variable names
- the names of child or parent components in the case of hierarchical components

The auxiliary information can be added for describing data such as grid type, resolution, physical units, and dimensions.

2.2 Data matching

Since the essential registered data are in the form of a string, data matching is string manipulation among registered components at all the levels. The output of data matching is the directional pair relationship from a producer component to a consumer component where at least the name of one export variable matches the name of one import variable. The matched data information can be used to reveal the ultimate source of exported variables.

2.3 Data filtering

Since an ESMF coupler is only used for coupling sibling components, the coupler will only be created between two components that have the same parent. So the hierarchical positions of components will be used to determine whether a coupler is created between two components that have the directional pair relationship revealed through data matching.

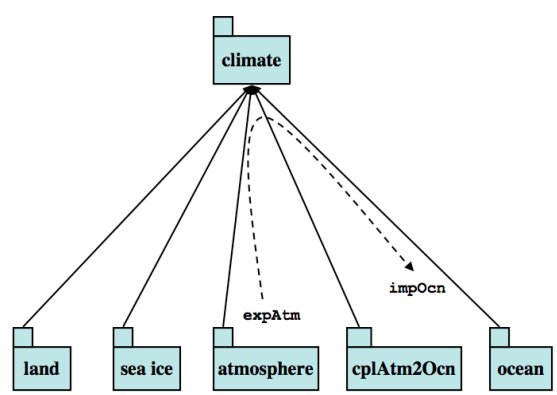

Figure 2. Illustration of coupling between two model components (e.g., atmosphere and ocean) through ESMF. The solid line designates the hierarchical relationship, while the dash line shows the data flow.

2.4 Code generator

Since the code structures and algorithms of model components and their transformation requirements (e.g., regridding) vary considerably from each other, we believe that generating the code for ESMF coupling is an effective way of minimizing the intrusion to the model code. This approach also allows a user to customize the coupling code to satisfy his or her specific requirements. So, our generic coupler toolkit will generate the stub code of ESMF components and couplers related to model coupling.

We chose to perform the transformation operation only at the coupler. So, the data transport from the source component of the matched export variable to the coupler can be done through redirecting the pointers. In that way, there is no need to include this variable (e.g., u or v) in the export state of the components along the path (e.g., sdyn in Fig. 4). This avoids the potential complexity of using nested ESMF states for coupling between multi-level hierarchical components. When there is a need for transformation such as regridding, a user must register the grid information associated with the import and export variables such as grid type and resolution. Before assigning the value of the export variable to its matched import variables, our generic coupler toolkit will examine whether the matched export and import variables have the same grid. If yes, the export variable will be assigned to the matched import variable directly. If not, a transformation operation will be carried out in the coupler.

3. Implementation

Since significant operations in data matching, data filter, and code generator are string manipulations, we chose C++ for implementation rather than Fortran.

For ease of use, the format of a plain text file with a simple rule serves an input. XML technology could be deployed if the complexity of input options increases considerably.

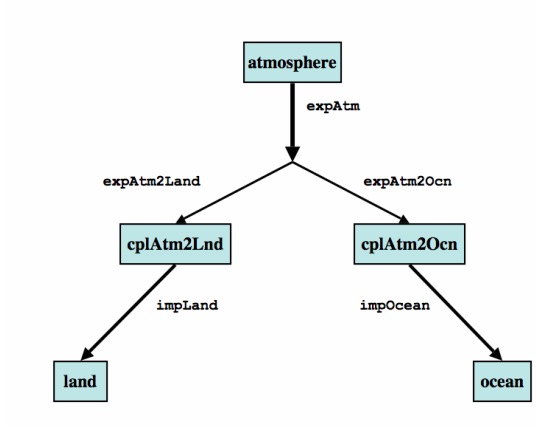

Figure 3. Illustration of a model component (e.g., atmosphere) coupling with two model components (e.g., land and ocean) through nested ESMF export states. The arrow designates the data flow.

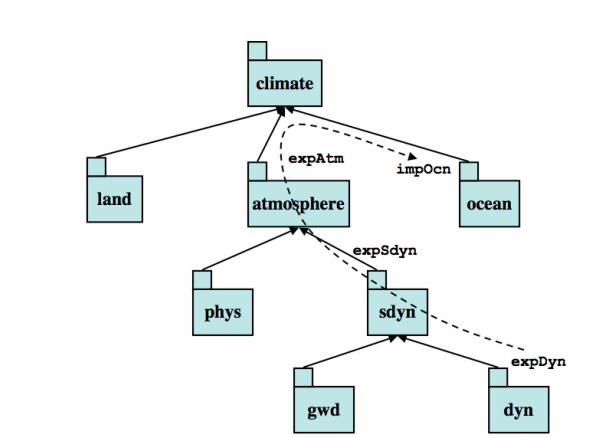

Figure 4. Illustration of coupling between two model components (e.g., dynamics and ocean) where there is a hierarchical relationship.

In this version, we implement the data matching, data filtering, and code generator in the case where there is no transformation and will add the features supporting transformation in the later version.

The output is the stub codes of ESFM components and couplers with the import and export states filled only with the matched variables. In addition, the directional export-to-import relations of components and the hierarchical component tree are written into text files, respectively, and can be visualized with the graph visualization software Graphviz [12].

These text files can also be generated automatically with a parser written in Perl. The parser traverses a user-specified directory tree, e.g. the src directory of GEOS-5, looking for valid gridded components and generates various lists containing information that includes import and export variable names and hierarchical component structure. The generated files are written in a format suitable for post-processing using Graphviz that allows on-the-fly graphical documentation depicting the system's components relationships.

4. Results

We tested our generic coupler toolkit with GEOS-5. One example of input files is the portion for the atmospheric component of GEOS-5, AGCM:

```
componentstart
AGCM
childstart
SDYN
PHYS
childend
parentstart
GCM
AANA
parentend
importstatestart
DUDT
DVDT
DTDT
```

```
DPEDT
DQVDT
DO3DT
importstateend
exportstatestart
DUDT_ANA
DVDT_ANA
DTDT_ANA
DPEDT_ANA
DQVDT_ANA
DO3DT_ANA
QVFILL
O3FILL
TROPP
TROPT
TROPQ
exportstateend
componentend
```

To add more components, a user can append the corresponding data into this file. Note that the names of components and import, and export variables should be consistent among all the registered components. In the case of cross-organization model component coupling, a naming convention such as CF is the likely choice, but our tool does not depend on it. It is the user's responsibility to establish the naming conversion inside his or her code. However, our generic coupler can be extended to support a name conversion if the user provides the name mapping between his or her local variable name and the corresponding standard name.

The portion of the generated code for AGCM component is as follows:

```
module AGCMMod
    use ESMF_Mod
    public AGCM_register
! Arrays
    public: TROPP, DPEDT, DTDT, DUDT,
      DVDT
! Fields
    type(ESMF_Field)::
      field_TROPP, field_DPEDT,
      field_DTDT, field_DUDT,
      field_DVDT

    contains
!-------------------------------------
    subroutine AGCM_register(comp, rc)

    type(ESMF_GridComp),
      intent(inout):: comp
    integer, intent(out) :: rc

    ESMF_GridCompSetEntryPoint(comp,
      ESMF_SETINIT,   AGCM_init1,
      1, rc)

    ESMF_GridCompSetEntryPoint(comp,
      ESMF_SETRUN, AGCM_run,
      ESMF_SINGLEPHASE, rc)

    ESMF_GridCompSetEntryPoint(comp,
      ESMF_SETFINAL, AGCM_final,
      ESMF_SINGLEPHASE, rc)
    end subroutine AGCM_register

    subroutine AGCM_init1(gcomp,
      importState, exportState,
      clock, rc)

    type(ESMF_GridComp),
      intent(inout):: gcomp
    type(ESMF_State), intent(inout)::
      importState, exportState
    type(ESMF_Clock), intent(in)::
      clock
    integer, intent(out) :: rc

    ESMF_StateAddField(importState,
      field_DPEDT, rc)
    ESMF_StateAddField(importState,
      field_DTDT, rc)
    ESMF_StateAddField(importState,
      field_DUDT, rc)
    ESMF_StateAddField(importState,
      field_DVDT, rc)
    ESMF_StateAddField(exportState,
      field_TROPP, rc)

    end subroutine AGCM_init1
```

The generic coupler toolkit can also be used to generate a diagram to reveal the hierarchical component relationship. Fig. 5 shows the hierarchical component tree of GEOS-5 generated based on the information of the components and their child components. Note that the arrow designates the hierarchical relationship. However, that relationship should be viewed from the perspective of "composition" rather than "inheritance" as supported in an Object Oriented Language such as C++ and Java.

Moreover, the generic coupler toolkit can create the export-to-import relations among all the registered components or the selected one with the rest of components. The corre-

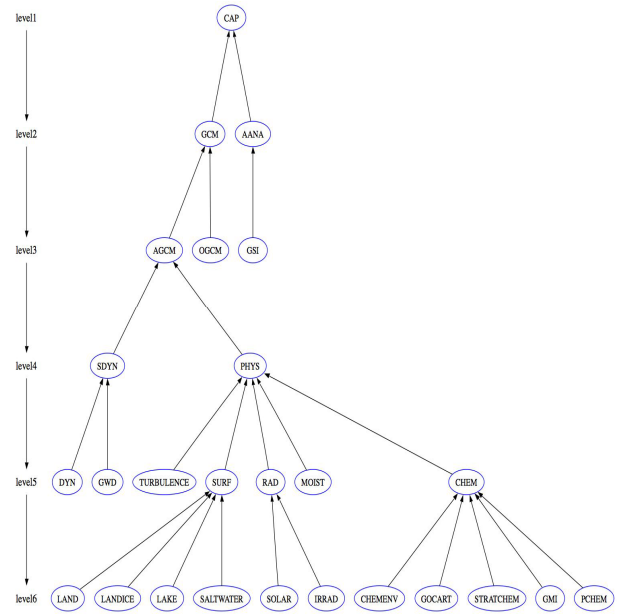

Figure 5. The diagram of the hierarchical component structure in GEOS-5.

Figure 6. The export-to-import relationships of the SURF component with others in GEOS-5. Note that the arrow designates the data flow from export variables to their matched import variables.

sponding matched variables are also written into a file for detailed analysis. Fig. 6 shows all the export-to-import relations of the SURF component with other GEOS-5 components. The line with an arrow connects the component with at least one matched name of export variable to the component with the matched import variable. This kind of relationship can be very helpful for an outsider to couple his or her components to GEOS-5 and understand how a new export variable impacts the scientific results of other components in GEOS-5.

Since the components of TURBULENCE, SURF, RAD, and MOIST are siblings (whose parent is PHYS) as shown in Fig. 5, the couplers should be used in the case of data exchange. As shown in Fig. 6, SURF has the matched export-to-import variables with the components of TURBULENCE, RAD, and MOIST. So the couplers are needed to exchange those variables. In the case that there is no need for transformation, the export variables can be passed to the matched import variables through reassigning the pointer.

5. Summary

As indicated in Figures 5 and 6, the coupling among hierarchical components can be very challenging. Our augmented design and implementation have addressed the issues in data matching and hierarchical component structure. However, further refinement is needed after more coupling applications are analyzed, in particular where transformation such as regridding is involved. It is challenging to balance standardization for facilitating model component coupling with the easy-to-adopt requirement since significant time and effort could be taken for those applications consisting of many components to adopt new standards. Nevertheless, cross-organization model coupling is the ongoing trend and can be fruitful scientifically and financially. Our generic coupler toolkit is developed to reduce the complexity for those models to adapt such a trend. Our initial testing with the production-quality climate code, GEOS-5, has shown that the generic coupler toolkit is helpful in understanding and facilitating the model coupling in a non-intrusive way.

Acknowledgments

This project is supported by the NASA Earth Science Technology Office's Computational Technologies Project. We would like to thank Joseph Spahr, Weiyu Yang, and Tom Clune for helpful discussions and Jarrett Cohen for editing the manuscript.

References

[1] S. Zhou, "Coupling Climate Models with Earth System Modeling Framework and Common Component Architecture," Concurrency Computation: Practice and Experience, 18 203 (2006).

[2] S. Zhou et al., "Cross-Organization Interoperability Experiments of Weather and Climate Models with the Earth System Modeling Framework," Concurrency and Computation: Practice and Experience, 19 583 (2007).

[3] L.C. McInnes, B.A. Allan, R. Armstrong, S.J. Benson, D.E. Bernholdt, T.L. Dahlgren, L.F. Diachin, M. Krishnan, J.A. Kohl, J.W. Larson, S. Lefantzi, J. Nieplocha, B. Norris, S.G. Parker, J. Ray, and S. Zhou, "Parallel PDE-Based Simulations Using the Common Component Architecture," an invited chapter in the book Numerical Solution of Partial Differential Equations on Parallel Computers, A. M. Bruaset, P. Bjorstad, and A. Tveito, editors, published by Springer-Verlag 2006.

[4] Common Component Architecture, http://www.cca-forum.org

[5] Earth System Model Framework, http://www.esmf.ucar.edu

[6] C. Hill, C. DeLuca, V. Balaji, M. Suarez, A. da Silva, and the ESMF Joint Specification Team, "The Architecture of the Earth System Modeling Framework," Computing in Science and Engineering, Volume 6, Number 1, 2004.

[7] Collins, N., G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva, " Design and Implementation of Components in the Earth System Modeling Framework," International Journal of High Performance Computing Applications, Fall/Winter 2005.

[8] S. Zhou and J. Spahr, "A Generic Coupler for Earth System Models," Proceedings of the Parallel CFD 2005 Conference, College Park, MD, U.S.A. (May 24-27, 2005) (Parallel Computational Fluid Dynamics: Theory and Applications, Edited by A. Deane et al., Elsevier B.V., page 187-193).

[9] The Weather Research and Forecasting (WRF) Model, http://www.wrf-model.org

[10] Joseph Spahr, private communication.

[11] The NASA Goddard Earth Observing System Model, Version 5 (GEOS-5), http://gmao.gsfc.nasa.gov/systems/geos5/index.php

[12] Graph visualization software (Graphviz), http://www.graphviz.org