International Journal of Engineering & Scientific Research
Vol. 7 Issue 1, January 2019,
ISSN: 2347-6532 Impact Factor: 6.660
Journal Homepage: http://esrjournal.com, Email: esrjeditor@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

COMPONENT COUPLING TECHNOLOGIES TO IMPROVE REUSE AND INCREASE EFFICIENCY OF INTEGRATED RESEARCH

Janne Umapathi[1], Dr. Vishal Khatri[2]
[1]Research Scholar, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.
[2]Research Guide, Department of Computing and Information Technology, Himalayan University, Itanagar, Arunachal Pradesh, INDIA.

ABSTRACT

Data, numerical models, and tools from a variety of sources are being combined by researchers all over the world to develop comprehensive models of the environment and related socioeconomic areas. Many other methods of linking these resources have been devised by academics. It is possible to choose from a wide variety of interfaces and frameworks, some of which require little or no code alterations, while others require a complete rewrite of your code. If we're all trying to connect data and models, why are we using so many different environments? One of the most basic premises is that different approaches can meet a variety of seemingly contradictory requirements such as generality and flexibility as well as usability, accuracy, and performance. Is each way better than the other? This study begins by looking into the common component architecture (CCA), earth system modelling framework (ESMF), FRAMES object modelling system (OMS) and OpenMI to solve these issues. Are they actually at conflict with each other, or do they complement each other to a significant degree

Keywords:integrated modeling, frameworks, interfaces, interoperability

1 Introduction

Ecological science comprises a broad range of branches, such as "meteorology, hydrology, geology, geomorphology, chemistry, and ecology," to provide the means to resolve the errands of today and tomorrow, with a different variation of prototypes from these and other fields, which must be blended. Interoperability was not taken into account when traditional (sub)mono-disciplinary numerical models were being developed. Many new challenges in information technology and semantics have emerged as a result of researchers from many disciplines working together, sharing data and models as well as models and tools. Diverse research communities around the world have looked into these issues, resulting in a range of behaviours, approaches, conventions, interfaces, and standardisation. Interoperability within these communities is becoming increasingly important as international cooperation grows (especially because of a smaller pool of researchers and similar research paradigms**(Gladwell, n.d.)**this, in turn, emphasises the importance of our interoperability). Having said that, what are the primary variations between the two?

As a result of their common goal of boosting reusability and reducing software development costs while also raising the overall efficacy of integrated research, the component coupling technologies investigated in this work appear to be competing and mutually exclusive..

Architecture: All generic aspects of a framework can be described in terms of architecture.

Component is a piece of software or a whole module. For the entire (or part of the) simulation domain, science components often represent a cohesive subset of physical processes

Environment is a set of "software services (infrastructure)" used to initiate, start, and end simulation components. Environment may or may not influence the communication between components.

Frameworkis a software architecture that can be reused in other projects. A runtime environment, support libraries, and components, as well as the interfaces and standards that go with them, are all included.

Interfaceis anabstract statement of the functions and methods it will use to communicate with other components at runtime.

ImplementationRealizing an architectural or an abstract component is what we mean by "implementation".

Coupling data is transferred sequentially or simultaneously between components. Memory or intermediate data files/repositories can be used to perform this type of data storage.

2 The players

Model coupling technologies developed by various groups are listed in alphabetical order in this chapter. Referenced articles and websites offer additional information.

2.1 CCA – Common Component Architecture

One of its goals is to build a "standard for high-performance component designs" that have HPC properties not accessible in existing generic components like CORBA and COM, as well as JavaBeans and.NET.

Components should be able to maintain their performance, and inter-component communication techniques should not be exclusive allow pre- and post-execution component configuration.

Language-specific component interface and ports for input and output arguments are defined in CCA using a scalar, an array or a function for the input and output parameters respectively. A framework must be in place to ensure compliance with CCA.

(1) First, SIDL can be used to construct the actual component interface wrappers,

(2) Components are instantiated and coupled via an API that can be configured,

(3) The API for accessing component repositories is also included in this release

(4)When it comes to plug-and-play interoperability between Fortran, C, C++, Java, or Python routines written in any of these languages (and many others), Babel tools are the standard**(Kumfert, 2003)**.

"The main CCA-compliant framework Ccaffeine", uses a command line tool and a graphical user interface. "CCA has been shown to be interoperable with ESMF and MCT", and the "CSDMS developershave successfully combined the Ccaffeine framework with the OpenMI 1.4 Java implementation. The CCA developments are currently led by TASCS3; this virtual organization isfunded through the SciDAC (Scientific Discovery through Advanced Computing) program of theUS Department of Energy funded through the SciDAC (Scientific Discovery through Advanced Computing) program of the US Department of Energy"**(Bernholdt et al., 2006)**

2.2 CHyMP – Community Hydrology Modeling Platform

When it comes to designing, providing, and supporting advanced simulation models for the academic community, the "Consortium of Universities for the Advancement of Hydrological Science (CUAHSI) relies on CHyMP4 Hydrology model components are the ultimate goal of this research, which aims to promote their development and adoption (contrary to the wide variety of smaller, incompatible and unsupported models that exist today). CHyMP and CSDMS' hydrology working group share several similarities."

A similar effort by the US National Weather Service (NWS) to establish a statewide early warning system using Delft-FEWS5 is not to be confused with this effort, which aims to develop the Community Hydrology Prediction System (CHPS). "One-way data streams and file-based data exchange" are the norm in operational systems like this one.

2.3 ESMF – Earth System Modeling Framework

All kinds of Earth science applications benefit from ESMF's high-performance architecture, including climate and numerical prediction. In order to facilitate parallelism, "ESMF" provides a coupling and utility framework that supports both MPI and open MP. Both infrastructure and superstructure are accessed by calls in the component code, which is tucked between the two. The utility layer includes all of these functions.

There are gridded and coupler components in the ESMF, which are distinct from each other (interpolation and mapping). The input and output arguments of the components are bundled together as input State and output state data structures, which must be defined by all components. An internal state can be used by components at their discretion. Arrays, (bundles of) Fields, and other states can be stored in these states.

An Array, which is a distributed, multidimensional array, can hold information such as the "array type, sort, rank, and halo widths". The "ESMF" Array structures can be used to cover existing Fortran/C arrays. A Field's data structure includes "data arrays, grid information, and metadata". Gridding and coupler components can be nestled within other gridding components that can be connected further up in the hierarchy. An executable that contains all of a drive module's components is not uncommon. There are UNIX, Linux, and Windows HPC platforms that support ESMF implementation. Recent advancements in the ESMF include integration with workflow management and visualisation services, the automatic development of couplers, executable and metadata, and online services. OpenMI and ESMF can be linked through web services. **(Saint & Murphy, 2010)**As of November 2009, the "NOAA Environmental Software Infrastructure and Interoperability (NESII) group", which receives funding from the "US Department of Defense (DoD), NASA, NSF, and the NOAA, has been home to the ESMF project."

2.4 FRAMES – Framework for Risk Analysis of Multi-Media Environmental Systems

"Models and tools (e.g., data retrieval and analysis) can communicate in the FRAMES" operational modelling environment for the US Environmental Protection Agency (EPA). To ensure that the components are correctly connected by end users, FRAMES uses well-established connection mechanisms and dictionaries. A total of 17 modules in 3MRA simulate waste stream toxins released from landfills and incinerators, as well as their fate and transit, exposure, and risk (both human and ecological) (e.g., landfills, waste piles). As a result of the large number of variables and processes, the model's conclusions are based on more than 10,000 simulations. While ESMF connects individual climate model components, 3MRA relies on simpler formulations for all of its domains. But there is an increasing need for more truthful portrayals.As a result, this is required for more sophisticated mode components and more complex component interaction. OpenMI will be the foundation for this new approach of interfacing**(Armstrong et al., 2009)**

2.5 HLA – High Level Architecture

For distributed real-time training and simulation scenarios, the "Defense Modeling and Simulation Office (DMSO)" developed the "High Level Architecture (HLA)" as a general-purpose architecture. In this topic, networks that are tightly coupled and exchange only a small amount of data are the norm. Finished in 1996, IEEE 1516 recognised it as a standard in 2000. It is possible to query a "simulation object model (SOM: description of component/federate and its possible exchanges") and a "federation object model (FOM: data exchanges occurring during simulation") using an "object model template (OMT)". It also specifies the basic architecture, the interfaces between components (federates), and the context in which they operate. The RTI makes it easier to exchange data. HLA is only a design; it doesn't do anything in the real world. As a result, existing HLARTI implementations are not 100% compliant with the IEEE standard for HLARTI interface implementations. It has been decided to define a DLC API (SISO-STD-004.1-2004) in order to assure uniformity in implementations. There should be no compatibility concerns with the new IEEE 1516-2010 standard, HLA Evolved IEEE 1516-2010.**(Craig et al., 2005)**

2.6 Kepler

The Kepler11 workflow architecture may be utilised in a wide range of scientific and technical domains since it allows "continuous time, discrete event, and dynamic or parallel data flow". The Java-based programme allows the user to see the workflow through the use of graphs of components ('actors'). The built-in actor library in Kepler contains 350 actors, which can be used for a variety of tasks, including numerical integration and image processing, as well as web service access and reading/writing standard file formats. The open source Kepler programme is being developed by a team from the "universities of Davis, Santa Barbara, and San Diego". A comparable Java-based workflow system was built by "Carole Goble of the University of Manchester" as part of the Grid project, which is currently part of OMII-UK.**(Gijsbers et al., 2010)**

2.7 MCT – Model Coupling Toolkit

"MCT13, an MPI-based library of Fortran90 modules", can be used to develop grid-based parallel integrated models (both structured and unstructured). For the CCSM3 coupler, version 2 of the toolkit was used, which was designed for and used by Local data in a 2D parameter-location array; it uses a Global Segment Map type to define the global division of a numerical grid in multiple operations. Local data is kept in an Attribute Vector type. Matrix-vector multiplication, spatial integration, and time averaging are used for intergrid interpolation in MCT, allowing for efficient parallel transmission of "MxN intercomponent data and MxM intracomponent data redistribution" depending on these data types. For sequential or concurrent execution, you can use MCT with a single executable system. MCT and cpl6 concepts will be applied to CCSM4 as well in ESMF.

2.8 OASIS and PALM

Prism is an initiative by ENES modellers to create a standard software architecture for their models. CERFACS was the driving force behind the development of the "open source OASIS3 and OASIS4" frameworks. The OASIS framework includes a driver, transformer, and "MPI-based PRISM System Model Interface Library (PSMILe)"**(Valcke & Morel, 2006)**explain, the PSMILe statically linked library is used for initialization, variable declaration, get and put operations as well as finalisation calls in the OASIS3 component executable. The driver component initialises and links the components during runtime based on the configuration files. When necessary, the data transformer can also be used to restructure and/or grid data; data that does not require this step can go directly through. CERFACS also created the PALM framework for oceanographic data assimilation applications for the MERCATOR project at the same time. MPI2 characteristics allow PALM to dynamically add and remove components during execution, unlike the OASIS couplers. But the parallel interpolation functionalities of OASIS4 are currently missing.**(Jacob et al., 2005)**

3 Comparing developments and conclusions

The end-user and collaborative aspects of FRAMES and CHyMP set them apart from the competition. Also, CSDMS, Delta Shell OpenWEB (locally) and iemHUB (online) are similar modelling environments**(David et al., 2013).**These initiatives, rather than inventing a new coupling technology, tend to accept (and adapt) existing ones; as a result, they are not included in this comparison. The other initiatives, in contrast to those of CCA, HLA, and OpenMI, have as their primary goal actual implementations, whereas these three initiatives are focused on architectures and interfaces. A reference implementation isn't provided by HLA's standard body; at least one is being developed by CCA and Open MI's developers.

These initiatives, rather than inventing a new coupling technology, tend to accept (and adapt) existing ones; as a result, they are not included in this comparison. The other initiatives, in contrast to those of CCA, HLA, and OpenMI, have as their primary goal actual implementations, whereas these three initiatives are focused on architectures and interfaces. A reference implementation isn't provided by HLA's standard body; at least one is being developed by CCA and Open MI's developers.

topic	CCA	ESMF	HLA	Kepler	MCT	OASIS	OMS	OpenMI	TIME
defines framework	✓	✓	✓	✓		✓	✓		✓
defines interfaces	✓	✓	✓		✓	✓	✓	✓	
provides (reference) implementation	r✓	✓		✓	✓	✓	✓	r✓	✓
defines object model		✓			✓	✓		✓	
code invasiveness	−−	+	?	?	?	?	++	−	?
plug & play (and graphical coupling)	✓		(✓)	✓				✓	✓
support for HPC environment	✓	✓	✓		✓	✓			
C/FORTRAN support	✓	✓	(✓)	W	✓	✓	W	W	W
Java support	✓		(✓)	✓			✓	✓	
.NET support								✓	✓

Table 1. Comparison of coupling technologies(David et al., 2013)

A graphical user interface (except for HLA) is provided by CCA, HLA, Kepler, OpenMI, and TIME for combining pre-compiled components from different developers at runtime. "ESMF, MCT, and OASIS target the high-performance user group with Fortran and C" as their principal programming languages. Fortran and its users are only partially supported by Kepler, OMS, OpenMI, and TIME, which all employ wrappers. The.NET platform is only supported by OpenMI and TIME. The implementation of HLA determines which languages are supported (generally Java or C). The only design that uses SIDL and Babel to handle the problem of language compatibility is CCA. Coupling components is becoming more common. Link validity and automatic link discovery depend on the use of ontologies and other metadata protocols. FRAMES and other operational frameworks already use some of these conventions. The ESMF and OASIS (component) developers collaborate with the US Earth System Curator28 and European METAFOR projects to implement Climate and Forecast standards.

As a result of this unifying vision, each of the systems presented below has tackled a distinct aspect of the integrated modelling challenge. A generic wrapper generator may be possible because of the commonalities in their interfaces (e.g., the Bespoke Framework Generator). Components might be moved across frameworks more easily if this were implemented**.**

Reference

Armstrong, C. W., Ford, R. W., & Riley, G. D. (2009). Coupling integrated Earth system model components with BFG2. *Concurrency and Computation: Practice and Experience*, *21*(6), 767–791.

Bernholdt, D. E., Allan, B. A., Armstrong, R., Bertrand, F., Chiu, K., Dahlgren, T. L., Damevski, K., Elwasif, W. R., Epperly, T. G. W., & Govindaraju, M. (2006). A component architecture for high-performance scientific computing. *The International Journal of High Performance Computing Applications*, *20*(2), 163–202.

Craig, A. P., Jacob, R., Kauffman, B., Bettge, T., Larson, J., Ong, E., Ding, C., & He, Y. (2005). CPL6: The new extensible, high performance parallel coupler for the Community Climate System Model. *The International Journal of High Performance Computing Applications*, *19*(3), 309–327.

David, O., Ascough II, J. C., Lloyd, W., Green, T. R., Rojas, K. W., Leavesley, G. H., & Ahuja, L. R. (2013). A software engineering perspective on environmental modeling framework design: The Object Modeling System. *Environmental Modelling & Software*, *39*, 201–213.

Gijsbers, P., Hummel, S., Vaneçek, S., Groos, J., Harper, A., Knapen, R., Gregersen, J., Schade, P., Antonello, A., & Donchyts, G. (2010). *From OpenMI 1.4 to 2.0*.

Gladwell, M. (n.d.). *The Tipping Point: How Little Things Can Make a Big Difference Irum Khattak*.

Jacob, R., Larson, J., & Ong, E. (2005). M× N communication and parallel interpolation in Community Climate System Model Version 3 using the model coupling toolkit. *The International Journal of High Performance Computing Applications*, *19*(3), 293–307.

Kumfert, G. (2003). *Understanding the CCA standard through Decaf*. Lawrence Livermore National Lab., CA (US).

Saint, K., & Murphy, S. (2010). *End-to-end workflows for coupled climate and hydrological modeling*.

Valcke, S., & Morel, T. (2006). *OASIS and PALM, the CERFACS couplers*. Tech. rep., CERFACS.