World Automation Congress © 2010 TSI Press.

TOWARDS AN INTELLIGENT SOFTWARE TOOL FOR ENHANCED MODEL INTEROPERABILITY IN CLIMATE CHANGE RESEARCH

SOHEI OKAMOTO[1], ERIC FRITZINGER[1], SERGIU M. DASCALU[1], FREDERICK C. HARRIS, JR.[1], SHAHRAM LATIFI[2], MICHAEL MCMAHON, JR.[1]

[1]*Department of Computer Science and Engineering*
University of Nevada, Reno
Reno, Nevada 89557, USA
{okamoto, ericf, dascalus, fredh, mcmahon}@cse.unr.edu
[2]*Department of Electrical and Computer Engineering*
University of Nevada, Las Vegas
Las Vegas, Nevada 89154, USA
shahram.latifi@cse.unr.edu

ABSTRACT— Model interoperability, also known as model coupling, is involved in a significant part of scientific research and offers an interesting and worthwhile area of exploration and development for related supporting software frameworks and environments. This paper presents the challenges of scientific model coupling, with emphasis on climate change research, and briefly describes the current main methods for dealing with model interoperability. The paper also proposes a design solution for a new intelligent software tool aimed at facilitating model coupling and improving research productivity in climate change research. The main components of the proposed tool are presented and the tool's major modes of operation are described. An outline of planned future work on the proposed software environment for enhanced model interoperability is also included in the paper.

Key Words: climate change research, model interoperability, model coupling, software tool, data portal.

1. INTRODUCTION

Climate change research has been mainly focused on the global scale [1, 2]. The research component of the new NSF EPSCoR-funded project aimed at enhancing Nevada's infrastructure for climate change science, education, and outreach is focused on providing new interdisciplinary capabilities to detect, analyze, and model the effects of regional and sub-regional climate change on atmosphere, landscapes, ecosystems, and water resources [3]. New models will be defined and used in connection with existing models to understand historical data, examine new data, generate views in different perspectives, and simulate new scenarios under various conditions. Data sets will range from observations such as temperature, precipitation, and aerosols to hydrologic and ecological data resources, and to records and statistics pertaining to demographic and health information systems [3, 4, 5, 6]. Models operating on such data are intertwined in the research analysis to understand the effects of climate change on ecosystem resources and support the use of newly acquired knowledge by policy makers [3, 4, 5, 6]. Currently, there are several other cyberinfrastructure-centered projects aimed at providing access to multidisciplinary datasets, including the National Ecological Observatory Network (NEON) [7], the Long Term Ecological Research Network (LTER) [8], and the Geosciences Network (GEON) [9].

A major challenge faced by the scientists is that climate change research models are not shared in an effective way due to heterogeneous (possibly incompatible) software solutions, lack of well-defined data descriptions, and the need of complex data transformations. For example, certain data points in a given format of a model' s output need to be extracted and transformed to a different data format expected by a subsequent model. The key problem in model interoperability is that the exchanged datasets need to be interpretable and interoperable between models. Currently, there are several ongoing projects aimed at overcoming heterogeneity in models and data formats and encodings by using various approaches, including standards for scientific high-performance component architectures [10], libraries of modules for integrated grid-based models [11], open interface standards for scientific modeling and simulation [12], high-performance frameworks for enhanced model interoperability in Earth science research [13], and persistent data structures [14].

Our NSF-funded project's cyberinfrastructure component is focused on the development of a Nevada Climate Change *data portal* that will provide to the project's research, education and policy making and outreach components a central point of access for various datasets and models [3]. This data portal will include as an important element a *software framework* aimed at enhancing model interoperability and thus increasing research effectiveness and productivity.

The fundamental problem of model and data interoperability is also rather common in other areas of research, which means that our proposed software framework (for now, given its early stage of development, referred to only as *software tool*) has the potential to be extended to other fields of scientific exploration where data conversions and model executions are regularly needed.

This paper describes our ongoing work on researching, designing and implementing this new software tool aimed at facilitating coupling and running various scientific models by people who are involved in climate change research, education, policy making, and outreach. In essence, the proposed software tool is designed to provide a user-friendly visual interface that supports three major modes of operation: model registration, scenario creation, and scenario execution.

The remainder of this paper is structured as follows: Section 2 provides additional background information on model coupling challenges and describes further our project's motivation and goals. Section 3 outlines our approach to develop the proposed software tool for enhanced model interoperability. Section 4 focuses on the tool's design and implementation principles and Section 5 presents details of its early prototype. Finally, Section 6 outlines directions of future work and concludes the paper.

2. BACKGROUND

The main goals of our Nevada-wide project are to develop a nationally recognized interdisciplinary program on climate change research, education, and outreach; to provide a virtual climate change center to detect, analyze, and model the effects of regional climate change on landscapes, ecosystems, and water resources; and to communicate research results to decision makers and the public [3]. There are six main project components (groups), as follows:

- Climate Modeling
- Ecological Change
- Water Resources
- Policy, Decision Making, and Outreach
- Cyberinfrastructure (CI)
- Education

Among these, the CI component (to which the authors of this paper belong) can be considered the "binding" component of the project, as it is meant to support and enhance communication and collaboration among all project groups. The CI group focuses on developing the Nevada Climate Change data portal, building an intelligent interactive software framework for increased model interoperability, and facilitating data sharing among project components.

The major difficulties in coupling models arise from heterogeneity and inconsistencies in software implementations, data formats, data structures, and data units, as well as from the need of complex data conversions to access and use the results of executing the models. Existing approaches of coupling models include *monolithic*, *scheduled*, *component*, and *communication* solutions [15].

3. APPROACH

Our proposed software tool follows the *scheduled* approach, meaning that the models are kept as separate programs (modules) and the output dataset of a model is used as input dataset for the subsequent model [15]. The key objective of our proposed software tool is to specify in an efficient and user-friendly manner the required connections between models (program modules) as well as the necessary transformations on the data exchanged by the models. The tool's design is characterized by a visual interactive user interface and a data flow model for model interconnection, data transfer, and model execution. The tool's main modes of operation are the following:

A. Model Registration

Models are registered (recorded and made available in the tool's execution environment) via associated descriptors that indicate the model's associated data and processes, properties, parameters, and other requirements for execution. Related data exchange information, such as input format, output format, and transformation rules are also defined using model descriptors. Such descriptors can be added to the system, edited (updated), and removed from the tool's environment.

B. Scenario Creation

The visual representation of models and model connections (using dedicated graphical symbols) is the distinguishing characteristic of the proposed software tool. Via the tool's user interface scenarios can be created (constructed) by connecting registered models (data and process model components) based on the models' interfaces and properties defined during their registration. A scenario is in essence a data workflow showing connected processes and the data exchanged between them. Specialized "data transformer" modules are also present in the tool's environment because they are necessary to specify data format conversions and other data transformations needed between the output of one model and the input of the next one. In a scenario, each model may have input connections from (and output connections to) more than one model, thus allowing the description of both sequential and parallel executions. Other data inputs and outputs to/from models include configuration files, databases, and other data stores. Scenarios can be saved, modified, and deleted.

C. Scenario Execution

A constructed scenario can be executed in the proposed software tool's environment by running its constituent processes and data transformations in the sequence specified by the scenario. While running a scenario the users are able to view the progress of the execution and obtain information on the current state of the models.

4. DESIGN AND IMPLEMENTATION

From the software architecture point of view the proposed tool follows a client-server model and consists of three layers. The first layer is the *presentation layer* through which the users can access the tool via the web-based data portal using their client computers and the available Internet connections. The users can remotely access the tool's visual interface to register models and create and run scenarios. A scenario's processes are run on the server at the tool's *application layer*, which has access and modification capabilities to the server's file system at the *data layer*.

The proposed software tool for increased model interoperability consists of several software subsystems, or packages (due to space limitations, only a brief description of these packages is provided here). Presentation capabilities and features are controlled by the tool's *GUI subsystem,* which uses the *Registration, Data, Process, Scenario,* and *Scenario Executor* packages. The *Scenario Executor* utilizes *Data*, *Process*, and *Scenario* subsystems, which are responsible for the application logic of the entire system. Lower level utility packages and functions are used by the tool's subsystems to provide the tool's functionality and capabilities.

5. PROTOTYPE SOFTWARE

Figure 1 shows the main window of the proposed software tool, in which an object panel is displayed on the left-hand side of the screen, and a blank scenario configuration panel on its right-hand side.

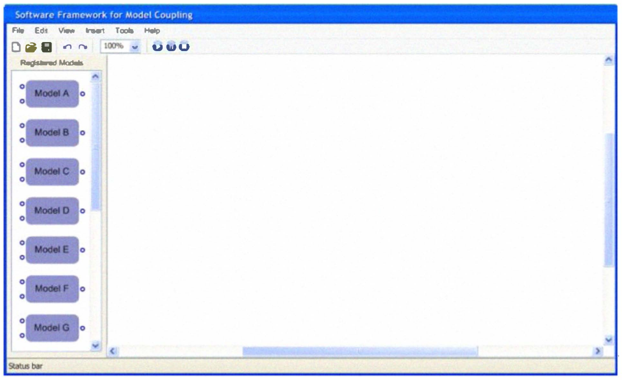

Figure 1: Main Window of the Proposed Software Tool for Model Coupling

Once the software tool's main window has been opened, the user may register new models (data and processes), update existing ones, or configure a scenario using available objects. The scenario can then be executed, and the user can make more modifications to the scenario, or go back to model registration.

Figure 2 shows the dialog window displayed during model registration (this includes editing the data and process components of the model or simply viewing them).

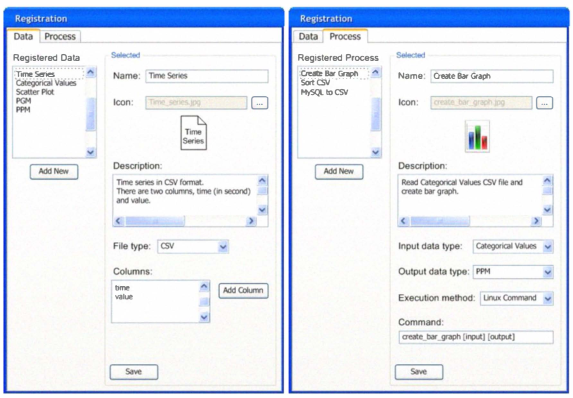

Figure 2: Model Registration Dialogs: Data and Process Components

The registration panel contains a data registration dialog under the “Data” tab (as shown on the left-hand side of Figure 2) as well as a process registration dialog under the “Process” tab (on the right-hand side of the figure).

In the scenario configuration panel (Figure 3), the user may select existing objects. Once an object is selected, the user can open and modify the properties of that particular object, and continue on to other objects.

Links between objects can be simply added by dragging the left mouse button from one connection point to another. The links between the objects provide a data flow to and from the objects during runtime. The scenario execution flows in a “left-to-right” manner, which means the connection points on the left-hand side of a model are for its input, and the connection points on the right-hand side of the model are for its output. Each process object determines how to handle the data as it is being received, and each such object determines how to send its data out.

During scenario execution the system will provide continuous updates on the progress and status of execution, using feedback from process executions when available. The system will check the link to the next process, wait for the completion of the current process, and proceed to the next process(es) based on the scenario’s defined dependencies.

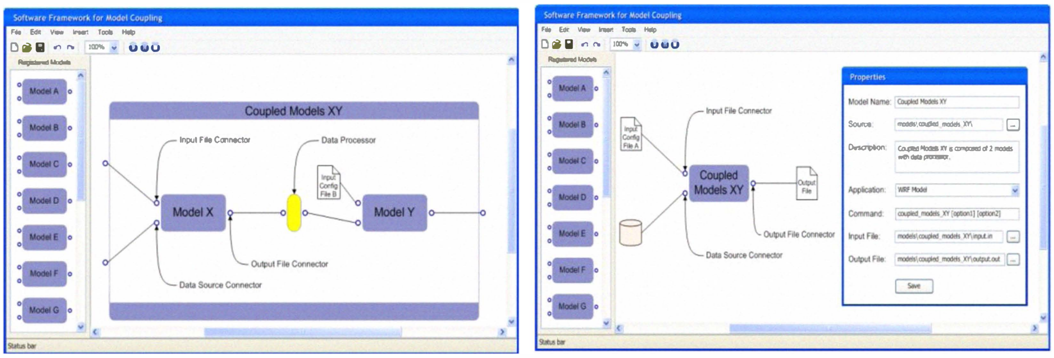

Figure 3: Software Tool for Model Coupling with a Customized Scenario

Figure 3 shows the view of creating a scenario that includes an object, Coupled Models XY, consisting of a pair of coupled models. The input and output files as well as the data source are linked with connectors to the processing object (the pair of coupled models). The properties dialog is also shown, where the user can specify additional details required for executing the scenario.

6. FUTURE WORK AND CONCLUSIONS

The first enhancement to the proposed tools will be to increase the tool’s functionality as well as its availability through a web interface, so that it can be accessed and run remotely. The latter will allow extended and easier access to the software tool and will facilitate its deployment, since future software updates could be performed directly on the server.

Another planned enhancement is to extend the software tool with features characteristic of an integrated development environment, where pieces of executable models could be developed via the tool’s interface. This will allow researchers and developers to build tailored executable software models at a single, central place.

In addition, since the software model development is primarily a collaborative work between researchers and software developers, an interface with capabilities for online collaboration needs to be included in the tool, where multiple people can contribute simultaneously to the construction of software models and their related data transfers and conversions.

Furthermore, the software framework will integrate an AI component to enhance the user's experience. This includes learning the tool's typical uses and the more frequent connections between models, thus becoming capable to offer suggestions to users for model and scenario development. The intelligent suggestion component will facilitate the users' work by simplifying their modeling decisions.

In this paper, we have described the background and motivation for a software tool aimed at enhancing model interoperability in climate change research, education, and outreach. Our approach is to design and develop a visual, dataflow-based tool that allows end-user configuration and execution of coupled models. The major modes of operation supported by the tool are model registration, scenario creation, and scenario execution, all available through a combination of form fill-in text entry and direct manipulation via the tool's user interface.

The early prototype of the software tool supports drag-and-drop actions that facilitate creating new scenarios that involve coupling models. The user can easily create scenarios by inserting model (process and data) objects and defining links and relationships between such objects.

Future work will focus on implementing more features and capabilities, exercising the tool on various modeling scenarios, providing a web-based interface to the tool, adding integrated development environment capabilities, and integrating an AI component that learns from the users' utilization of the tool and guides the scenario creation and execution activities.

REFERENCES

1. US Global Change Research Program, http://www.globalchange.gov/
2. 2009 Western Consortium of Idaho, Nevada and New Mexico http://www.webs.uidaho.edu/epscor/Web%20Pictures/Western_Consortium.htm
3. Nevada EPSCoR: Nevada Infrastructure for Climate Change Science, Education, and Outreach, http://www.nevada.edu/epscor/nsf-overview.html
4. New Mexico EPSCoR, http://www.nmepscor.org
5. Idaho EPSCoR, http://www.webs.uidaho.edu/epscor/
6. Climate & Global Change Dynamics - Climate Change Research, http://www.cgd.ucar.edu/
7. A. Zimmerman and B.A. Nardi, "Whither or whether HCI: requirements analysis for multi-sited, multi-user cyberinfrastructures," *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, ACM, 2006, pp. 1601-1606.
8. H. Karasti, K.S. Baker, and E. Halkola, "Enriching the Notion of Data Curation in E-Science: Data Managing and Information Infrastructuring in the Long Term Ecological Research (LTER) Network," *Comput. Supported Coop. Work*, Kluwer Academic Publishers, 2006, Vol. 15(4), pp. 321-358.
9. M. Gahegan, J. Luo, S.D. Weaver, W. Pike, and T. Banchuen, "Connecting GEON: Making sense of the myriad resources, researchers and concepts that comprise a geoscience cyberinfrastructure," *Computers & Geosciences*, 2009, Vol. 35(4), pp. 836-854.
10. Common Component Architecture Forum, http://www.cca-forum.org/
11. Model Coupling Toolkit, http://www.mcs.anl.gov/research/projects/mct/
12. OpenMI Association, http://www.openmi.org/reloaded/
13. Earth System Modeling Framework, http://www.earthsystemmodeling.org/
14. J.S. Horsburgh, D.G. Tarboton, M. Piasecki, D.R. Maidment, I. Zaslavsky, D. Valentine, and T. Whitenack, "An integrated system for publishing environmental observations data," *Environmental Modelling & Software*, 2009, Vol. 24(8), pp. 879-888.
15. T. Bulatewicz, Support for Model Coupling: An Interface-based Approach. *PhD thesis*. University of Oregon, June 2006.