

Towards Decentralized Integration of Environmental Resources on the Web: A Resource Alignment Service

Peishi Jiang^a, Mostafa Elag^a, Luigi Marini^b, Liu Rui^b, Praveen Kumar^{a,*}

^a*Ven Te Chow, Hydrosystem Laboratory, Civil and Environmental Engineering, University of Illinois, Urbana, IL, USA*

^b*National Center for Supercomputing Applications, University of Illinois, Urbana, IL, USA*

Abstract

...

Keywords: Model Web, semantic interoperability, the Resource Alignment Service (RAS)

1. Introduction

For decades, integrated environmental modeling has been applied in investigating the mechanism of the entire ecosystem, which becomes increasingly complex under climate change and human impact recently (Argent, 2004; Laniak et al., 2013). With the growing efforts in integrated modeling, scientists with a solo discipline background are able to access, reuse and couple resources (i.e., models and data) from various disciplines. It therefore enhances their assessment, prediction and management abilities of a complicated natural system. One of the approaches is through envisioning a world of "Model Web", where the functionalities of a model are exposed through web services and then integrated via the internet (Geller & Turner, 2007; Geller & Melton, 2008; Nativi et al., 2013). As stated by Nativi et al. (2013), one of the challenges in achieving the web-serviced model integration is to ensure the 'minimal barriers to entry' or the interoperability between the serviced resources. In this study, we aim to envision a world of 'Model Web' with emphasis on ensuring the semantic consistency of information flowing between web-serviced resources in geoscience.

The application of service-oriented architecture in environmental modeling just starts recently (Castronova et al., 2013; Mineter et al., 2003; Granell et al., 2010; Goodall et al., 2011, 2013). Following the concept of Software as a Service (SaaS), models and data can also be exposed as web services, called as Model as a Service (MaaS) and Data as a Service (DaaS), respectively. To achieve a world of interoperable MaaS and DaaS, the concept of "Model Web" was put forward by (Geller & Melton, 2008). Later, Nativi et al. (2013) explicitly defined it as "a dynamic web of models, integrated with databases and websites, to form a consultative infrastructure where researchers, managers, policy makers, and the general public can go to gain insight into what if questions" in initiating it under the project of the Group on Earth Observation. Several efforts have been made to draw the picture of Model Web. Goodall et al. (2011) exposed water resource models as web services through the Open Geospatial Consortium (OGC) Web Processing Service (WPS) and demonstrated

*Corresponding author

Email address: kumar1@illinois.edu (Praveen Kumar)

how it could be encapsulated as OpenMI(open modeling interface and environment)-compliant models. Then, they furthered the idea of servicing an OpenMI-compliant model based on WPS by considering the case of time-dependent models (Castronova et al., 2013). Furthermore, an OpenMI-ESMF web service wrapper was developed to couple a climate model implemented via ESMF web service with an OpenMI-compliant hydrologic model running on a personal computer (Goodall et al., 2013).

One of the key challenges in model integration community is the heterogeneity or long-tail property of models and data. Due to different habits of research groups or individual scientists, models developed by them would probably differ in operating system, programming language, variable name, data unit, temporal-spatial property and so forth. In the world of Model Web, a user of a web-serviced model does not need to consider the operating system or programming language of which the model is serviced and implemented because he or she only interacts with the exposed service. But the user is required to be aware of both the communication protocol the service applies and the remaining heterogeneities (e.g., variable name, data units). Otherwise, the semantic inconsistency of information flowing between services would impact the functionality of a web service negatively as shown in 1. To achieve a seamless integration of serviced resources, a uniform protocol and an aligning mechanism are required for information processing and ensuring reducing the semantic consistency of data flowing between serviced resources, respectively. The existing efforts described in the previous paragraph try to achieve such interoperability by applying a specific standard. For instance, apply OpenMI standard in integrating water resource models serviced by WPS (Goodall et al., 2011; Castronova et al., 2013). Nevertheless, the application of a certain standard or framework might not be fruitful. This is because the existence of different frameworks in different domains might imply the adoption of a specific framework in one discipline, thus complicating the online communication of serviced resources from various disciplines. For example, an OpenMI-compliant serviced model cannot talk with a serviced model embedded in Object Modeling System (OMS, David et al. (2013)) without a proper wrapper for communication between OpenMI and OMS. Furthermore, standards like OpenMI do not allow semantic mediation between variable names, prohibiting the integration of models with variables in alias names.

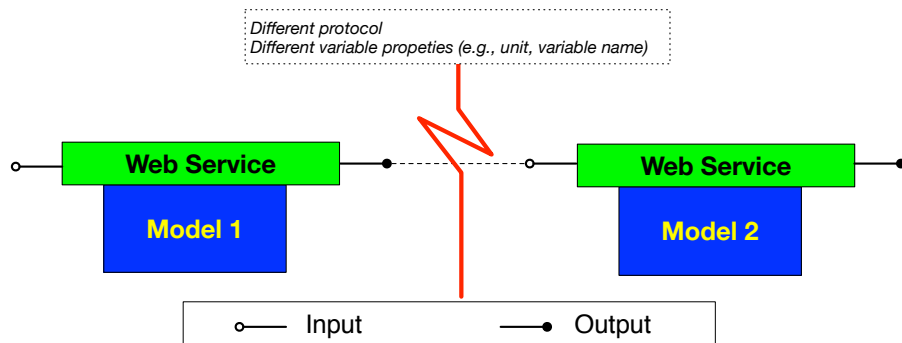


Figure 1: An illustration of the semantic inconsistency between two web serviced models.

To tackle the issue of semantic interoperability of integrating service-oriented modeling, we develop a web

service named Resource Alignment Service (RAS) to ensure semantic consistency of information exchanged
 50 between MaaS over the web, including: (1) semantic mediation between variable names; (2) conversion of
 mismatched units; (3) temporal alignment of resource time horizon and (4) spatial alignment of resources
 spatial attributes. Models are implemented by the WPS, which exposes both its execution and its inputs
 and outputs (I/O) properties through web service. Then, RAS is utilized to align the quantities exchanging
 between MaaS over the web the two types of serviced models.

55 In the remainder of the paper, Section 2 describes the design requirement and provides a short overview
 on model integration community. In Section 3, the architecture of the Resource Alignment Service (RAS is
 presented, including the an brief overview of GeoSemantic framework, the functionality and the components
 of RAS. Section 4 demonstrates the application of RAS in coupling WPS-implemented models with the input
 data from Clowder, an online data repository. First we create a workflow of heterogeneous collection of data
 60 and WPS-implemented models. Then, we show how RAS can seamlessly align the semantics of quantities
 exchanging between these resources. Finally, a brief summary is given in Section 5, with mentions on the
 future work.

2. Design Requirement and Overview

Modeling integration in geoscience has advanced significantly these years because of the increasing collab-
 65 oration between scientists or research groups from various disciplines (e.g., hydrology, geology, ecology and so
 on). The traditional integrated modeling approach is through tight coupling, where the initial independent
 models are coupled under a modeling integration application or framework. The representative integration
 frameworks include the Community Surface Dynamics Modeling System (CSDMS), Earth System Modeling
 Framework (ESMF), OpenMI standard and so forth (Hill et al., 2004; Peckham et al., 2013; Moore & Tindall,
 70 2005). The CSDMS architecture employs the component-based modeling approach, where standand-alone
 models are encapsulated and couplped with others in a plug-and-play manner. For each model, a BMI (Basic
 Model Interface) file is required for recording the model’s basic information and converting it into a plug-
 and-play component. Then, the integration process is conducted in the Component Model Interface based
 on all the utilities provided by CSDMS frameowork. ESMF is aimed at coupling models from Earth science
 75 (e.g., wearther and climate model), with a ‘sandwich’ pattern characterizing its architecture (Hill et al.,
 2004). The user code layer contains all the model components in ESMF, which are then encapsulated in the
 superstructure layer aligning and connecting the input and output data between models. All the utilities for
 the alignment are provided in the infrastructure layer. In addition, OpenMI is an effort on model coupling in
 water resource community. It provides a standard interface for any new or existing models exchanging data
 80 with each other during running period. Now the OpenMI standard is adopted as a specification in the OGC.

Nevertheless, despite the advantage of the tight-coupling approach that the framework entirely controls
 both the data structures and processes of all parts of the model, it is difficult to couple models not following the
 standard or convention due to the fixed convention or standard of a framework. On the contrary, through the

service-oriented loose-coupling approach, modelers only need to focus on the standardization of web interface and the protocol of data exchanging (Goodall et al., 2011). As shown in 2, by using web technology, the functionality of a model can be exposed as a web service with a specific protocol for data communication. In this case, by being conserved in its own hardward environment, any model requiring large data consumption or complicated configuration process can be easily coupled into any workflow or big decission system. It then also allows individual model developers or groups to maintain and update their models conveniently.

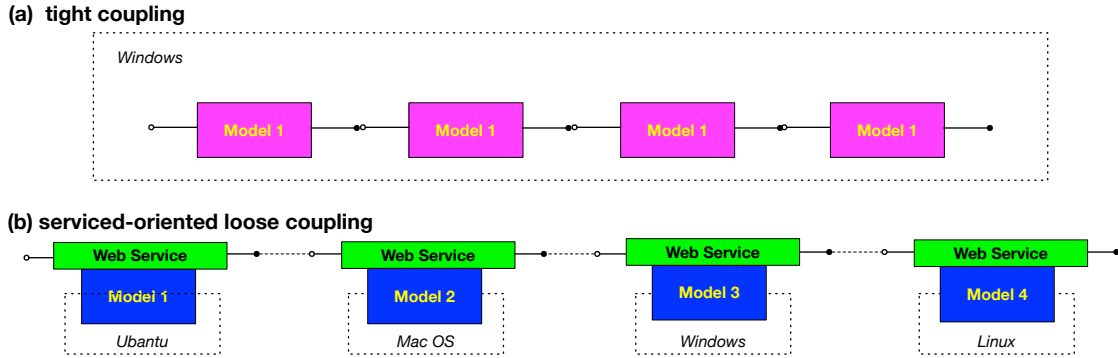


Figure 2: A comparison between the tight-coupling approach and the serviced-oriented loose-coupling approach: (a) all the models are linked under the same framework and operated in the same hard environment; (b) the models implemented under various hard environment talk with each other by exposing their functionalities through web services.

Therefore, MaaS or DaaS, being able to exchange data with other serviced models over the web, requires a uniform communication protocol for information exchanging. In geoscience, the applied protocols include Water Markup Language (WaterML) and the Open Geospatial Consortium (OGC) standards for transmitting hydrologic time series data and geospatial data, respectively. WaterML is initially applied as the protocol of hydrologic data transmission in WaterOneFlow web services held by Consortium of Universities for the Advancement of Hydrologic Scienc, Inc - Hydrologic Information System (CUAHSI-HIS, Valentine et al. (2007)). Now the second version of WaterML is adopted as one of the OGC standards. Furthermore, various types of standards are published by the OGC, which includes but not limited to: (1) specifications for data format such as geography markup language (GML) and keyhole markup language (KML); (2) specifications for data publication such as web coverage service (WCS), web mapping service (WMS) and web feature service (WFS); and (3) specifications for geoprocessing such as web processing service (WPS). Swain et al. (2015) provides a good review of the current open source solutions for water resources web application, most of which apply various OGC standards. In this study, as a geoprocessing standard, OGC WPS is implemented by the web serviced model, which has proven to be successfully applied in other modeling studies requiring data exchanging between services (Castronova et al., 2013; Goodall et al., 2011; Schaeffer, 2008; Vitolo et al., 2012).

The OGC WPS provides an approach for standardizing data processing, including three methods: GetCapabilities, DescribeProcess and Execute. The GetCapabilities method allows clients to receive a list of

existing processes or models from the remote server implementing the WPS. The DescribeProcess method replies clients with the metadata of the inputs and outputs of a specific model or process. Finally, the execution of a model or process can be conducted through the Execute method based on the input values from clients. All the three methods can be requested by either HTTP POST or HTTP GET method. In this study, models are implemented by WPS, and we utilized the DescribeProcess and Execute method for retrieving the metadata specifics of the models and the model execution, respectively.

References

- Argent, R. M. (2004). An overview of model integration for environmental applications components, frameworks and semantics. *Environmental Modelling & Software*, 19, 219–234.
- Castronova, A. M., Goodall, J. L., & Elag, M. M. (2013). Models as web services using the open geospatial consortium (ogc) web processing service (wps) standard. *Environmental Modelling & Software*, 41, 72–83.
- David, O., Ascough, J., Lloyd, W., Green, T., Rojas, K., Leavesley, G., & Ahuja, L. (2013). A software engineering perspective on environmental modeling framework design: The object modeling system. *Environmental Modelling & Software*, 39, 201–213.
- Geller, G. N., & Melton, F. (2008). Looking forward: Applying an ecological model web to assess impacts of climate change. *Biodiversity*, 9, 79–83.
- Geller, G. N., & Turner, W. (2007). The model web: a concept for ecological forecasting. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International* (pp. 2469–2472). IEEE.
- Goodall, J. L., Robinson, B. F., & Castronova, A. M. (2011). Modeling water resource systems using a service-oriented computing paradigm. *Environmental Modelling & Software*, 26, 573–582.
- Goodall, J. L., Saint, K. D., Ercan, M. B., Briley, L. J., Murphy, S., You, H., DeLuca, C., & Rood, R. B. (2013). Coupling climate and hydrological models: Interoperability through web services. *Environmental Modelling & Software*, 46, 250–259.
- Granell, C., Díaz, L., & Gould, M. (2010). Service-oriented applications for environmental models: Reusable geospatial services. *Environmental Modelling & Software*, 25, 182–198.
- Hill, C., DeLuca, C., Suarez, M., Da Silva, A. et al. (2004). The architecture of the earth system modeling framework. *Computing in Science & Engineering*, 6, 18–28.
- Laniak, G. F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M. et al. (2013). Integrated environmental modeling: a vision and roadmap for the future. *Environmental Modelling & Software*, 39, 3–23.

Mineter, M. J., Jarvis, C., & Dowers, S. (2003). From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. *Environmental Modelling & Software*, 18, 379–391.

Moore, R. V., & Tindall, C. I. (2005). An overview of the open modelling interface and environment (the openmi). *Environmental Science & Policy*, 8, 279–286.

Nativi, S., Mazzetti, P., & Geller, G. N. (2013). Environmental model access and interoperability: The geo model web initiative. *Environmental Modelling & Software*, 39, 214–228.

Peckham, S. D., Hutton, E. W., & Norris, B. (2013). A component-based approach to integrated modeling in the geosciences: The design of csdms. *Computers & Geosciences*, 53, 3–12.

Schaeffer, B. (2008). Towards a transactional web processing service. *Proceedings of the GI-Days, Münster*, .

Swain, N. R., Latu, K., Christensen, S. D., Jones, N. L., Nelson, E. J., Ames, D. P., & Williams, G. P. (2015). A review of open source software solutions for developing water resources web applications. *Environmental Modelling & Software*, 67, 108–117.

Valentine, D., Zaslavsky, I., Whitenack, T., & Maidment, D. R. (2007). Design and implementation of cuahsi waterml and wateroneflow web services. In *Proceedings of the Geoinformatics 2007 Conference, San Diego, California* (pp. 5–3).

Vitolo, C., Buytaert, W., & Reusser, D. (2012). Hydrological models as web services: An implementation using ogc standards. In *Proceedings of the 10th International Conference on Hydroinformatics, HIC, Hamburg, Germany*.