IN13B-1841

WMT: The CSDMS Web Modeling Tool

Mark Piper, Eric Hutton, Irina Overeem, and James Syvitski

Community Surface Dynamics Modeling System, University of Colorado Boulder
mark.piper@colorado.edu

What is CSDMS?

The Community Surface Dynamics Modeling System (CSDMS [1]) deals with the Earth's surface—the dynamic interface between lithosphere, hydrosphere, cryosphere, and atmosphere. We are an international community that promotes the modeling of earth surface processes by developing and disseminating software modules that predict the movement of fluids, and the flux of sediment and solutes in landscapes and their sedimentary basins. CSDMS strives to expand the use of quantitative modeling techniques, promotes best practices in coding, and advocates for the use of open-source software.

The CSDMS modeling framework

CSDMS has developed a framework that allows models to be coupled and executed in a high-performance computing (HPC) environment [2]. The framework uses tools from the Common Component Architecture (CCA [3]) toolchain, including Babel, which provides interoperability between models written in C, C++, Fortran, Java, and Python, and Bocca, which creates and manages CCA-compliant components. To include a model in the CSDMS framework, a developer must implement a Basic Model Interface (BMI [4]), which includes methods to initialize and finalize the model, update the model state, and get and set model state variables. Implementing a BMI is nonintrusive, but it may require refactoring for some models.

What is WMT?

To streamline and standardize access to models, CSDMS has developed the Web Modeling Tool (WMT), a RESTful web application with a client-side graphical interface and a server-side database and API that allows users to build coupled models in a web browser on a personal computer or a mobile device, and run them in an HPC environment.

With WMT, users can:

- Design a one-way or a coupled model from a set of components
- Edit component parameters
- Save models to a web-accessible server
- Share saved models with the community
- Submit runs to an HPC system
- Download simulation results

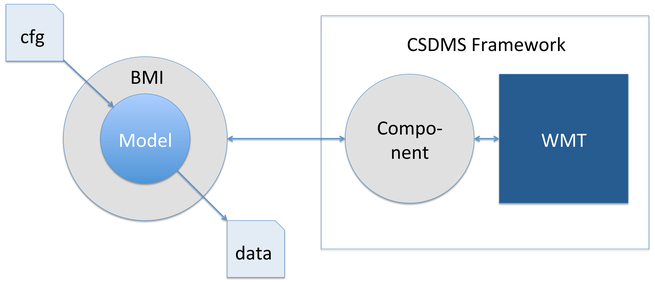

Figure 1. The architecture of CSDMS. A model wrapped with a Basic Model Interface (BMI) can be wrapped with Babel to become a component in the CSDMS Framework. Components use BMI methods to communicate with other components, which may be written in C, C++, Fortran, Java, or Python. WMT puts a standardized graphical interface on a component, and provides an environment where components can be coupled and run on a high-performance computing cluster.

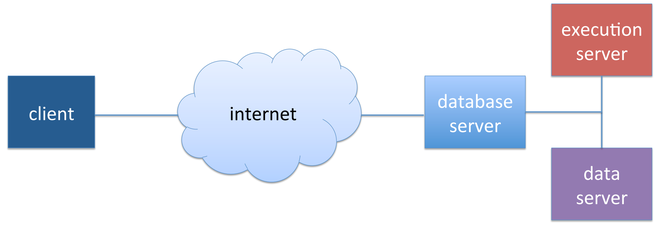

Figure 2. The architecture of WMT. In a RESTful application, the client and server are separated. The client (what the user sees) stores little information, making it portable. The server (where the work is done) isn't concerned with actions in the client, making it scalable. The client and server communicate through HTTP GET and POST methods, using JSON to carry information. WMT is stateless: each message passed between the client and the server contains all the information needed to perform an action.

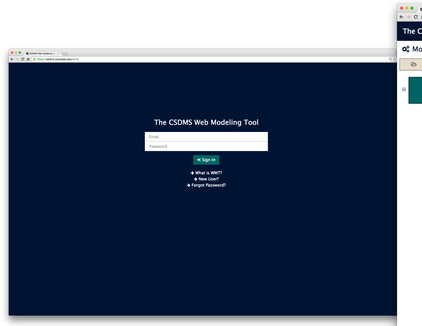

Figure 3. The WMT sign-in screen.

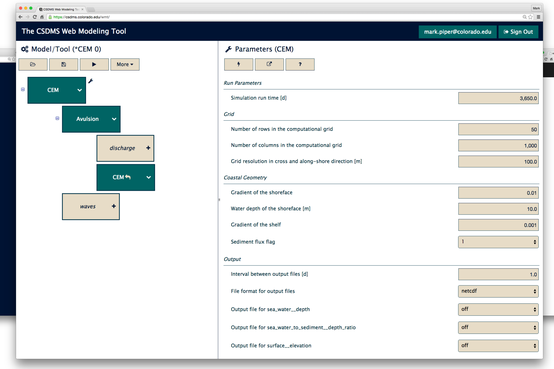

Figure 4. The main screen of the WMT client. A user can choose from a list of available components to couple, configure the component's parameters, save the coupled model, and run it on an HPC system.

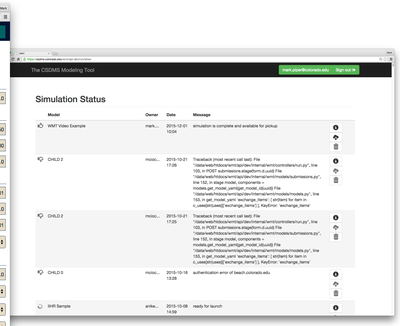

Figure 5. The WMT simulation status screen.

What is WMT? (cont.)

The **WMT client** [5] is an Ajax application written in Java with the GWT toolkit. For deployment on the web, the GWT compiler translates Java code to optimized and obfuscated JavaScript. The WMT client is supported on Firefox, Chrome, Safari, and Internet Explorer.

The **WMT server**, written in Python and SQLite, is a layered system, with each layer exposing a web service API:

- *wmt-db*: database of component, model, and simulation metadata and output [6]
- *wmt-api*: configure and connect components [7]
- *wmt-exe*: launch simulations on remote execution servers [8]

The database server provides, as JSON-encoded messages, the metadata for users to couple model components, including descriptions of component exchange items, uses and provides ports, and input parameters. Execution servers are network-accessible computational resources, ranging from HPC systems to desktop computers, containing the CSDMS software stack for running a simulation. Once a simulation completes, its output, in NetCDF, is packaged and uploaded to a data server where it is stored and from which a user can download it as a single compressed archive file.

References

1. CSDMS: http://csdms.colorado.edu
2. James P M Syvitski, Eric W H Hutton, Mark D Piper, Irina Overeem, Albert J Kettner, and Scott D Peckham. Plug and play component modeling - the CSDMS 2.0 approach. In *Proceedings of the 7th Intl. Congress on Env. Modelling and Software*, International Environmental Modelling and Software Society (iEMSs). San Diego, CA, 2014
3. CCA: https://computation.llnl.gov/casc/components
4. Scott D Peckham, Eric W H Hutton, and Boyana Norris. A component-based approach to integrated modeling in the geosciences: The design of CSDMS. *Computers & Geosciences*, 2012.
5. WMT-client: https://github.com/csdms/wmt-client
6. WMT-db: https://github.com/csdms/wmt
7. WMT-api: https://github.com/csdms/wmt-api
8. WMT-exe: https://github.com/csdms/wmt-exe

Acknowledgments

The CSDMS Integration Facility operates under continuing grant 0621695 from the US National Science Foundation.