The Journal of Open Source Software

The Basic Model Interface 2.0: A standard interface for coupling numerical models in the geosciences

Eric W.H. Hutton[1], **Mark D. Piper**[1], **and Gregory E. Tucker**[1, 2, 3]

1 Community Surface Dynamics Modeling System, University of Colorado Boulder **2** Cooperative Institute for Research in Environmental Sciences (CIRES), University of Colorado Boulder **3** Department of Geological Sciences, University of Colorado Boulder

DOI: 10.21105/joss.02317

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @yangbai90
- @teuben

Submitted: 29 May 2020
Published: 21 July 2020

Summary

Component modeling is a research technique in which new models are constructed by coupling the inputs and outputs of simpler existing models. Component modeling traces its roots to component-based software engineering, where a software system is constructed from a number of independent, reusable software components, each encapsulating a unit of functionality and exposing inputs and outputs through an interface. A tangible analogy is a bicycle. A bicycle is a system of reusable, replaceable components. Tires are one of the components. You can easily swap in a studded tire for icy winter streets, then swap it out again in the summer.

While there is a longer history of component modeling in fields such as climate modeling, with, for example, the Earth System Modeling Framework (Collins et al., 2005), component modeling is relatively new to the earth surface processes community. Some recent examples include Ratliff, Hutton, & Murray (2018), who show that a river model transporting sediment can feed a delta model that distributes the sediment, and Hoch, Eilander, Ikeuchi, Baart, & Winsemius (2019), who show that coupling hydrologic and hydrodynamic models may sharpen inundation estimates in flood modeling.

In component-based software engineering, components communicate through interfaces: named sets of functions with prescribed arguments and return values. The bicycle analogy above benefits from a standard interface for tire diameter and width. Likewise, component modeling can benefit from an interface for describing the inputs, outputs, and behaviors of a model. The *Basic Model Interface* (BMI) provides a standard set of functions for querying, modifying, and running a model. Equipping a model with a BMI allows the model to be coupled with other models that expose a BMI. The BMI concept was introduced by Peckham, Hutton, & Norris (2013) as a foundational technology for the proposed Community Surface Dynamics Modeling System (CSDMS) model coupling framework. The current work represents an update to the original BMI, with new functions for describing variables and for working with structured and unstructured grids. Full documentation for the current version of the BMI is available at https://bmi.readthedocs.io.

The functions that comprise the BMI are designed to be straightforward to implement in any programming language, using only simple data types from standard language libraries. To generalize across languages, the BMI is expressed in the Scientific Interface Definition Language (Epperly et al., 2012). BMI specifications for four languages—C, C++, Fortran, and Python—have been derived from the SIDL specification. For each language, links to the GitHub repositories supplying the specification and an example implementation are listed in Table 1 below. Detailed instructions for building the language specification and example are given within each repository.

Table 1: Repositories containing BMI language specifications and examples. Prefix the CSDMS GitHub organization (https://github.com/csdms/) to the repository name to obtain the full repository URL.

Language	Specification	Example implementation
C	bmi-c	bmi-example-c
C++	bmi-cxx	bmi-example-cxx
Fortran	bmi-fortran	bmi-example-fortran
Python	bmi-python	bmi-example-python

While CSDMS currently supports the four languages listed in Table 1, a BMI can be created for any language. BMI is a community-driven standard; contributions that follow the contributor code of conduct listed in the main BMI repository are welcomed, and are acknowledged in the repository and the documentation.

Acknowledgements

This work is supported by the National Science Foundation under Grant No. 1831623, *Community Facility Support: The Community Surface Dynamics Modeling System (CSDMS)*.

References

Collins, N., Theurich, G., Deluca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., et al. (2005). Design and implementation of components in the earth system modeling framework. *The International Journal of High Performance Computing Applications*, *19*(3), 341–350.

Epperly, T. G., Kumfert, G., Dahlgren, T., Ebner, D., Leek, J., Prantl, A., & Kohn, S. (2012). High-performance language interoperability for scientific computing through babel. *The International Journal of High Performance Computing Applications*, *26*(3), 260–274.

Hoch, J. M., Eilander, D., Ikeuchi, H., Baart, F., & Winsemius, H. C. (2019). Evaluating the impact of model complexity on flood wave propagation and inundation extent with a hydrologic-hydrodynamic model coupling framework. *Natural Hazards and Earth System Sciences*, *19*(8), 1723–1735.

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.

Ratliff, K. M., Hutton, E. H. W., & Murray, A. B. (2018). Exploring wave and sea-level rise effects on delta morphodynamics with a coupled river-ocean model. *Journal of Geophysical Research: Earth Surface*, *123*(11), 2887–2900. doi:10.1029/2018JF004757