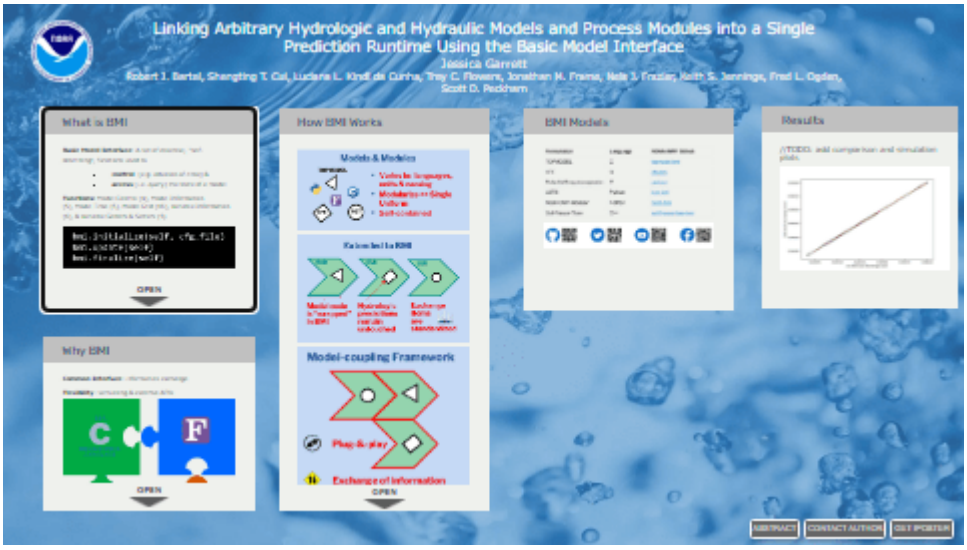
Linking Arbitrary Hydrologic and Hydraulic Models and Process Modules into a Single Prediction Runtime Using the Basic Model Interface



Jessica Garrett

Robert J. Bartel, Shengting T. Cui, Luciana L. Kindl da Cunha, Trey C. Flowers, Jonathan M. Frame, Nels J. Frazier, Ahmad Jan, Keith S. Jennings, Fred L. Ogden, Scott D. Peckham

PRESENTED AT:

Frontiers in Hydrology

WHAT IS BMI

Basic Model Interface: A set of essential, "self-describing", functions used to

- **control** (e.g. advance-of-time) &
- **access** (i.e. query) the state of a model

Functions: Model Control (4), Model Information (5), Model Time (5), Model Grid (16), Variable Information (6), & Variable Getters & Setters (5)

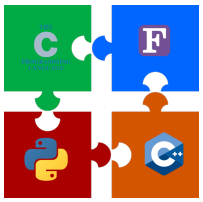
```
bmi.initialize(self, cfg_file)
bmi.update(self)
bmi.finalize(self)
```

Developed by: **Community Surface Dynamics Modeling System (CSDMS (https://csdms.colorado.edu/wiki/Main_Page))** Wiki (https://csdms.colorado.edu/wiki/BMI) | Docs (https://bmi.readthedocs.io/en/latest/) | GitHub (https://github.com/csdms/bmi)

WHY BMI

Common Interface: information exchange

Flexibility: versioning & external APIs

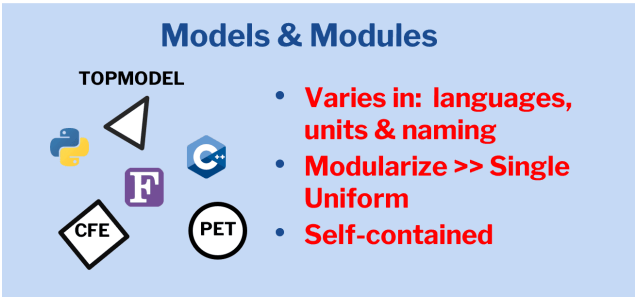


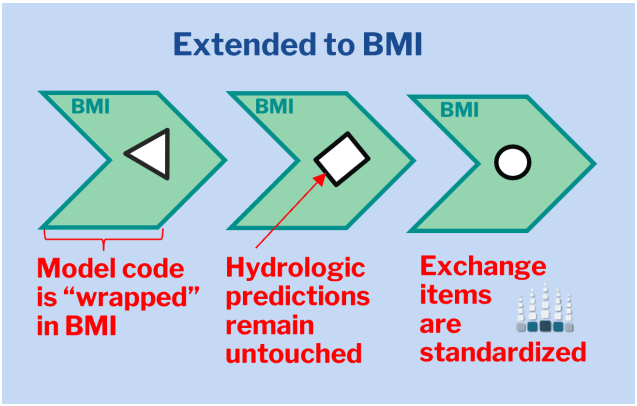
Interoperability: language agnostic (C, C++, Python, Fortran & Java) & extendable as a data component (https://csdms.colorado.edu/wiki/DataComponents)

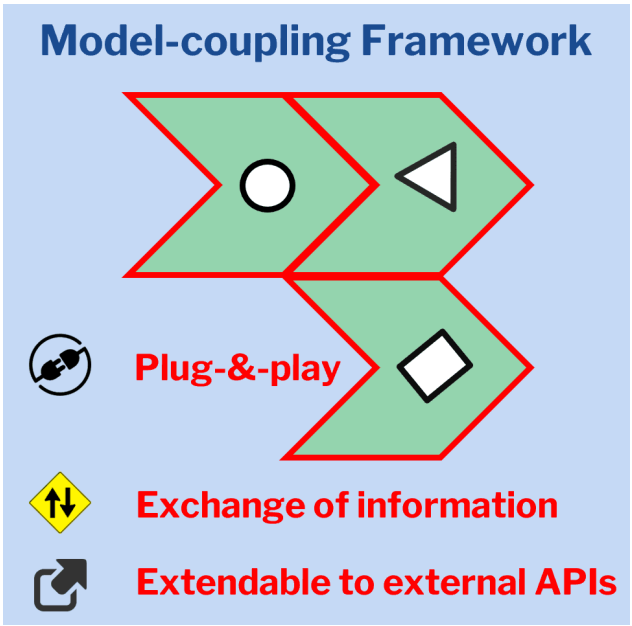
Non-invasive: predictions unchanged & runtime performance is minimally affected

Community: names (CSDMS (https://csdms.colorado.edu/wiki/CSDMS_Standard_Names)) & units (UDUNITS (https://www.unidata.ucar.edu/software/udunits/)) are standardized & simple implementation

HOW BMI WORKS







ADAPTED MODELS & MODULES

Formulation	Language	Github.com/NOAA-OWP/
TOPMODEL	C	topmodel
CFE	C	cfe
Potential Evapotranspiration	C	evapotranspiration
Long-**S**hort-**T**erm-**M**emory	Python	lstm
Noah-OWP-Modular	Fortran	noah-mp-modular
Soil-Freeze-Thaw	C++	SoilFreezeThaw
Soil-Moisture-Profile	C++	SoilMoistureProfiles
Snow17*	Fortran	snow17
LGAR*	Python	alt-modular/tree/GAR/Modules/GAR

(*) Denotes models and modules still under BMI development




jessica.garrett@noaa.gov water.noaa.gov

ADAPTION NOTES

```c
// ********************************************************************
// ************** BMI: MODEL CONTROL FUNCTIONS *************
// ********************************************************************

static int Initialize (Bmi *self, const char *cfg_file)
{
    topmodel_model *topmodel;
    topmodel = (topmodel_model *) self->data;

    // Read and setup data from file
    init_config(cfg_file, topmodel);

    // Initialize model variables
    topmodel->current_time_step=0;
    topmodel->sump = 0.0;
    topmodel->sumae = 0.0;
    topmodel->sumq = 0.0;

    return BMI_SUCCESS;
}

static int Update (Bmi *self)
{
    topmodel_model *topmodel;
    topmodel = (topmodel_model *) self->data;

    double current_time, end_time;
    self->get_current_time(self, &current_time);
    self->get_end_time(self, &end_time);
    if (current_time >= end_time) {
        return BMI_FAILURE;
    };

    topmodel->current_time_step += topmodel->dt;

    // call model run (1 timestep)
    topmod(topmodel->output_fptr,topmodel->nstep, topmodel->num_topodex_values, ...

    return BMI_SUCCESS;
}

static int Finalize (Bmi *self)
{
  if (self){
    topmodel_model* model = (topmodel_model *)(self->data);

    water_balance(model->output_fptr, model->yes_print_output, ...

    results(topmodel->output_fptr, topmodel->out_hyd_fptr, topmodel->nstep, ...

    free(self->data);
  }
    return BMI_SUCCESS;
}
```

Results

Runtime Performance
- Minimal computation cost
- Expected values – results unchanged

Model Code:
- Minimal adjustments to primary function definitions
- Maintains original internal names and units
- Still recognizable by original author/developer?

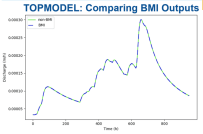
Control Functions

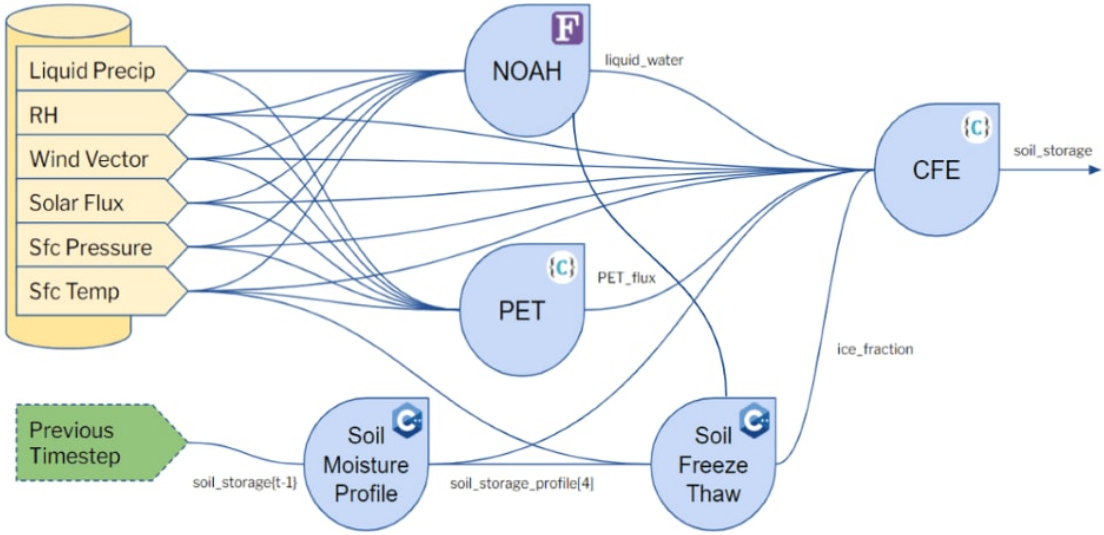
bmi.initialize():
- **Reads configuration**
- Open/close files
- Allocates memory
- Sets initial data values

bmi.update():
- Advances model state
- Iterates clocktime

bmi.finalize():
- Frees memory
- Objective functions

TOPMODEL: Comparing BMI Outputs





ABSTRACT

The National Water Model (NWM) currently uses a single land surface model (LSM), NOAH-MP, to support operational streamflow predictions in the United States. Previous research has indicated that a single uniform hydrologic model would likely misrepresent the water cycle across large space and time extents because of uncertainties in model parameters and inherent process variability. Therefore, using a modular modeling system to simulate varied stormflow processes offers the potential to improve overall hydrologic predictions. However, many existing modular infrastructures require scripting of subcomponents in the same language as the model's core driver, hindering incorporation of new routines. Similarly, such systems typically demand shared variables to also express distinct identical properties; i.e., common names, data types and units across all modules. This suggests increased adaptability and modifiability by minimizing such qualitative requirements.

Developed by the Community Surface Dynamics Modeling System (CSMDS) group, the Basic Model Interface (BMI) standard enables such flexibility. *Basic* speaks to the most fundamental and universal components of computational modeling. *Interface* refers to a space where often otherwise self-contained methods exist and communicate freely. When a hydrologic model is extended to BMI, a set of newly defined *self-describing* functions allow access and control the model's state. Notably, these functions act as thin middleware and do not interfere with a model's core functionality; i.e., the equations and solvers used in the code to represent components of the hydrologic cycle remain unaltered. When implemented into a framework, BMI allows information exchange from one model or module to another. For example, a control program can pass output from a BMI-enabled potential evapotranspiration routine to a BMI-enabled stormflow module to remove evaporated and transpired water from the soil matrix. Models adapted include TOPMODEL, the Conceptual Functional Equivalent (CFE) of the current operational National Water Model, and components of the NOAH-MP land surface model. We will discuss the steps needed to adapt a model to BMI along with its relative benefits, robust multi-language support, and runtime advantages in the NextGen infrastructure.