

Status update on BMI core and extensions

Bert Jagers

bert.jagers@deltares.nl

January 20, 2023

Table of contents

- BMI: Basic Model Interface
- BMI governance
- BMI extensions / changes being discussed
- BMI at Deltares
- Demand for parallel coupling
- Towards BMI for parallel applications

BMI: Basic Model Interface



- <https://github.com/csdms/bmi>, <https://csdms.colorado.edu/wiki/BMI>
- **Leading developer:**
U Colorado, Community Surface Dynamics Modeling System group (since 2007)
- **Current version:** 2.0 <https://doi.org/10.21105/joss.02317>
- **Focus:** *model interface*
different frameworks/drivers using BMI exist, e.g. PyMT (<https://github.com/csdms/pymt>).
- **Main design principle:**
A BMI-compliant model does not make any calls to the infrastructure and is not modified to use data structures of the infrastructure. "Hollywood principle"
 - The model exposes functions for initialization, time stepping, finalization, getting/setting data together with meta data describing the data

BMI: Basic Model Interface

- initialize
- update, update_until
- finalize

- get_component_name
- get_input_item_count, get_input_var_names
- get_output_item_count, get_output_var_names

- get_var_type, get_var_units, get_var_nbytes
- get_var_grid, get_var_location

- get_start_time, get_current_time, get_end_time
- get_time_units, get_time_step

- get_value, get_value_ptr, get_value_at_indices
- set_value, set_value_at_indices

- get_grid_rank, get_grid_size, get_grid_type, get_grid_shape, get_grid_spacing, get_grid_origin
- get_grid_x, get_grid_y, get_grid_z
- get_grid_node_count, get_grid_edge_count, get_grid_face_count, get_grid_edge_nodes, get_grid_face_edges, get_grid_face_nodes, get_grid_nodes_per_face

BMI governance

- 2022: Introduction of the “BMI Council” to discuss developments.
- Current members represent:
 - U Colorado
 - NOAA NWS-NWC
 - USGS
 - Deltares
 - NL eScience Center
 - TU Delft

BMI extensions / changes being discussed

- Identifying model state (data assimilation)
 - dedicated call or introduce a purpose flag with all quantities
- eXtended Model Interface
 - <https://www.sciencedirect.com/science/article/pii/S1364815221002991>
 - extension of BMI
 - communication per iterations
- Distinction between core and extensions
 - Versioning
 - Listing of extensions supported by a component
- New language bindings (current list: Fortran, C, C++, Java, Python)
 - Julia, MATLAB, ...
- Coordinate reference systems (not just models in global coordinates)
- HPC parallel interface

BMI at Deltares

- Delft3D Flexible Mesh
 - D-Flow FM – unstructured flow solver: 1D, 2D, 3D
 - D-Morphology – sediment transport and bed update (integrated in D-Flow FM)
 - D-Waves – multi-grid wave (using SWAN)
 - D-Water Quality – water quality and ecology processes, optionally integrated in D-Flow FM
 - D-Real Time Control – “human” control of barriers, structures, etc.
 - DIMR – Deltares Integrated Model Runner
- WFLOW – hydrological model, Cartesian (Julia)
- MODFLOW (collab USGS) – ground water unstructured
- XBeach – flow, morph, wave – storm event
- SFINCS – compound flooding, Quadtree, GPU
- WANDA – pressurized pipeline systems, 1D



Demand for parallel coupling

- D-Flow FM with D-Waves/SWAN – coastal morphodynamic development
- D-Flow FM with MODFLOW 6 and MetaSWAP (unsat zone) – surface/ground water
- XBeach/D-Flow FM with AEOLIS (wind driven transp), vegetation and MODFLOW – beach and dune zone development
- D-Flow FM with vegetation dynamics (Python) – long-term delta development
- D-Flow FM with PALM (atmospheric) – effect of water bodies on urban thermodynamics
- D-Flow FM with WHIFFLE (atmospheric) and D-Waves – local ocean model for wind farm scale
- D-Flow FM – integration as coastal module in NOAA NWS NextGen CONUS hydrofabric*

Clear need to use a standardized approach ... but not all applications may use the same coupling infrastructure

Approach

- Define BMI interface for MPI environment
- Implement this BMI extension for our components
- Determine infrastructure used per application
 - For “in-house” applications most likely OASIS or ESMF
 - However, need for portability between Linux and Windows.
- Add an infrastructure specific cap on top of BMI



Towards BMI for parallel applications

Inspired by interfaces of existing HPC infrastructures in ESM domain (OASIS, ESMF, ...) and beyond (e.g. preCICE; <https://precice.org/>)

1. Support passing MPI communicator to component
 - As part of Initialize call – modifies the standard BMI
 - Before the Initialize call – risk with fuzzy state
2. Support querying component for partitioning of “meshes” (location sets)
 - Fairly well defined for the main mesh(es)
 - Output at special locations: “hydraulic structures”, “observation points” and “observation cross-sections” (multiple partitions may contribute to the same cross-section: expose or hide?)
 - Output of particles/drifters: distribution over threads is dynamic
3. Query component for input/output quantities per mesh
4. Get/set data in line with spatial partitioning

Contact

 www.deltares.nl

 [@deltares](https://twitter.com/deltares)

 [linkedin.com/company/deltares](https://www.linkedin.com/company/deltares)

 info@deltares.nl

 [@deltares](https://www.instagram.com/deltares)

 [facebook.com/deltaresNL](https://www.facebook.com/deltaresNL)



Deltares