OpenMI: A standard interface for linking environmental models

Costas Gavardinas[1], Fotis Fotopoulos[1], Peter Gijsbers[2] and Roger Moore[3]

Abstract

Environmental processes and interactions have always been a modeling challenge not only because of their complexity but also because they usually demand an integrated analysis that can be achieved only by interlinking domain-specific and case-tailored models. However, the scientific community has reached a general consensus that no single model or modeling system can represent adequately the full range of these processes. Moreover, up to now the flexibility of such linkages has been minimal and entirely dependent on modeling software developers. The HarmonIT project, a European Union funded project bringing together developers from all over Europe, has tried to solve this problem by designing and implementing a standard interface for linking mathematical models: the Open Modeling Interface & Environment (OpenMI).

Although work on the OpenMI initiated in the water management domain, the architecture that has been developed is highly abstract and can accommodate the linking of modeling software in several other domains. Furthermore data visualization, real-time monitors and numerical data sources (e.g. databases) can be linked in seamlessly. This flexibility is achieved by realizing the model linkage at the computational engine level of the model and regarding it as a data-accepting and providing entity; the OpenMI is essentially a pull-based pipe and filter architecture consisting of communicating components. Using this data exchange paradigm, the engine is accessed directly by other linked components during model run time.

One of the main requirements during the design of the OpenMI was that migrating legacy models should be a relatively straightforward procedure, as only this would ensure the environment's usefulness and viability. Indeed, the OpenMI architecture imposes specifications on the way and the format of the data being exchanged. However the model engine itself normally requires minimal changes in order to support the OpenMI component interface. Once the engine has become a Linkable Component (in OpenMI terminology), coupling it with other Linkable Components is a simple procedure. After the link setup, data exchange is triggered by a data request to the component at the end of the linked models chain.

In order to assure the sanity of the exchanged data, it must be expressed in a commonly understood format (a linked component "lingua franca"). The OpenMI defines the base data (numerical values) and its semantics in terms of quantity (what), element set (where), time (when) and data operations (how). Furthermore, meta-data is used to specify the data that *can* or *will* actually be exchanged. The actual data transfer is realized by calling a single method of the component: *GetValues()*. This may require additional computations, data-operations (unit change, interpolation, extrapolation …) or even further requests for data to other linked components.

Additional mechanisms have been put in place to simplify the linked simulation setup and execution: the OpenMI architecture specifies XML descriptors for locating, accessing and initializing the binary software

[1] National Technical University of Athens, School of Civil Engineering, Department of Water Resources, Hydraulic and Maritime Engineering, Laboratory of Hydrology and Water Resources Management, Iroon Polytechniou 5, 157 80 Athens, Greece,
Tel. +30 210 772 2878, Fax. +30 210 772 2879, E-mail: {cgavar,ffotop}@chi.civil.ntua.gr

[2] WL|Delft Hydraulics, P.O. Box 177, 2600 MH Delft, The Netherlands, E-mail: peter.gijsbers@wldelft.nl

[3] Centre for Ecology & Hydrology, Wallingford, Oxfordshire, OX10 8BB, United Kingdom, E-mail: rvm@ceh.ac.uk

unit (model executable). Moreover, a lightweight event support enables the passing of messages between components and allows for call stack tracing, progress monitoring, fault handling or even communicating with components not directly involved in the simulation (e.g. visualization and output tools, external controllers, etc.). Finally exceptions are defined and are thrown when irrecoverable errors occur.

In order for a model to become OpenMI-compliant few adjustments have to be made that will allow direct access to its engine. Firstly, the model must be able to expose information about the variables it can input or output and their locations (if applicable). The model initialization must be separate from the actual computation phase and boundary conditions should be collected during computation. The engine should be always able to provide a value for the requested variable, time and location, either by direct computation, data operation or delegated query to another linked component. Missing values in a value set should be flagged and finally inability to compute an entire result set should lead to an exception.

The OpenMI interface specification is defined in the org.OpenMI.Standard namespace, with its default implementation being the org.OpenMI.Backbone namespace. Although sufficient on their own for model linkage, these namespaces do not deal with practical issues such as the model engine migration, the setup of linked model simulations and their deployment. The engine migration may entail the need for additional utilities (buffering, aggregation, spatial and temporal operations), whose development is not trivial. In order to facilitate the transition to a wrapped model, OpenMI provides a software toolbox that may prove invaluable to modelers and model coders. The *org.OpenMI.Utilities* namespace includes a generic model wrapper, available in several common programming languages that can "talk" directly, via an internal interface, to the existing engine core. A buffer is also provided and has been extended with data operations (unit conversion, temporal aggregations). To complement the wrapping facilities, a default implementation for spatial conversions is being provided. In addition, an AdvancedControl package has been created to assist in directing iterations and calibration. To enable reuse (define, save, load, modify) of component chains, a Configuration package is incorporated which supports deployment as well. To exploit all these facilities, a front-end has been developed in the org.OpenMI.Tools namespace. This package provides visual facilities for editing components chains, event logging and straightforward data visualization. Note however, that no one is forced to use the OpenMI environment as provided, as long as they implement the OpenMI interfaces in a sound way.

It is apparent that the applicability of the OpenMI architecture is not limited to water domains but can easily accommodate socio-economic, ecological, biological, chemical, geological or any other interactions, as long as they can be modeled mathematically. Once a model has been migrated into OpenMI, it can become a node in any linked simulation, given it understands the exchanged quantities and the semantics of the links are valid. This will be demonstrated by the migration of two models TLWaterNET, a 1-D pressure water distribution network and TLSewerNET, a 1-D sewer network, and their application for coupling the water and sewer networks of the city of Amfissa, Greece. According to common practice, the water network is designed first, and then a percentage of each node demand is manually transferred to the sewer network pipes. However, there are many limitations to this procedure. First of all, the base demand in a water network varies with time. This implies that when a percentage of a node demand is transferred to the sewer network, one has to choose a single value from a wide range of inflows. Secondly, if one or more properties of the water network are changed, a new manual solution of the sewer network is required which can be a lengthy task and a source of errors. Finally, it is not possible to study the real-time interaction of the two networks or replace a model with another, without an excessive amount of work.

Last but not least, the OpenMI is an open standard: making a model OpenMI-compliant does not require licensing or other copyright limitations. Moreover, having already achieved the support of leading commercial software developers as well as academics from around the world, the OpenMI is expected to be supported, maintained and improved well into the future by an OpenMI consortium. Hopefully the OpenMI environment will relieve the environment modeling and management community of the physical linking problem and will enable it to focus on other aspects of integrated systems such as their use as decision support systems.