Harpham, Quillon; Cleverley, Paul

Integrated environmental modelling with OPENMI and Fluidearth

Hydrolink

INTEGRATED ENVIRONMENTAL WITH OPENMI AND FLUIDEARTH

BY QUILLON HARPHAM AND PAUL CLEVERLEY

Integrated Environmental Modelling
It is becoming increasingly recognised that many modern environmental questions cannot be answered by simulating physical, chemical or biological phenomenon in isolation. Environmental systems typically involve a variety of natural processes working together. The environment is an interconnected system and multiple, dependent environmental phenomenon may interact resulting in consequences which were not expected if each had been considered independently. If we wish to use numerical models to simulate these processes accurately then our modelling methods must take into account their interactions as well as accurately simulating each individually. The phenomena must be allowed to influence one another's behaviour. For example, a flood may be a combination of groundwater and surface water. At some places, the groundwater feeds the surface water; at others the surface water feeds the groundwater. We need to understand both together to correctly predict the outcome. This being the case, the only way to successfully answer these questions is to employ integrated approaches which allow this to happen. These will often span disciplines, to complement the traditional single discipline methods.
A number of approaches exist for achieving this. One is to embody the interactions of all relevant phenomena in a single 'super-model', that is, create a single numerical modelling application which incorporates all necessary processes. However, this can quickly produce an application which becomes unwieldy, difficult to develop and support and ultimately vulnerable by its dependence on certain key individuals. Indeed, it is becoming clear that one single numerical model cannot be sufficient to represent all of the details needed for decision making and planning.
Another approach is to simulate complex systems by integrating multiple, smaller models that collectively simulate the larger problem in question, that is, to build an integrated composition of previously independent numerical models and run them together. In its simplest form this can be achieved by taking the output of one model and using it as the input to another. Such a 'one-way' interaction has been common in environmental modelling for many years. Requirements have since developed to demand a more flexible, interoperable and extensible solution. Moreover, a 'two-way' connection between numerical models is often required. Both models need to be given the opportunity to influence each other. It is desirable for each component to remain sufficiently independent so that experts can remain in their disciplines, yet are able to communicate model outputs clearly where necessary at the interfaces between their coupled models. Indeed, achieving this in a standardised fashion will better enable easy extensibility of the integrated composition to incorporate new parameters and to exchange similar numerical engines where appropriate.

The OpenMI Standard
OpenMI is a software component interface standard for the computational core of a numerical model. It was developed by leading hydraulic centres across Europe as part of projects part funded by the European Commission as a response to EU Water Framework Directive calls for integrated water management. As such, it was originally developed as a means for coupling existing models which would typically consider the interactions of environmental processes, in particular involving water. The computational core (or engine) of a numerical model is designed or adapted to be 'OpenMI Compliant'. Such compliant components can then be put together in OpenMI integrated compositions. This would typically occur between two components running simultaneously through time-steps which span a time horizon. They would then send and/or receive data at specific time-steps as each proceeds through its respective time interval. In this way, the two model components can both influence the results produced by the other at each point where data is exchanged. The linked components may run asynchronously with respect to these time-steps or proceed through together. OpenMI also supports one-way passing of data from a driving component to a second, set up only to receive. The latest version of OpenMI, version 2.0, was released in December 2010 at a specially convened reception during an EU-US summit in Washington DC. A short history of OpenMI is available on the OpenMI Association website (1).

The FluidEarth Implementation of OpenMI
OpenMI can be represented on paper as a set of object interface specifications. In order to save software developers having to create these interfaces from the paper definitions, the OpenMI Association publishes a set of open source reference interfaces in C# and Java on Source Forge (2) and encourages developers to use these.
In addition to the standard itself, the OpenMI Association pledges to accompany each release of OpenMI with two tools:

- A Software Development Kit (SDK) allowing numerical model developers to more easily make their model engines OpenMI compliant;
- A Graphical User Interface (GUI) allowing numerical model users to build and run compositions of OpenMI compliant components.

HR Wallingford's FluidEarth 2 is an implementation of OpenMI 2.0 for Windows .Net 4.0 consisting of a set of such tools which provide an environment for the standard to be used. FluidEarth began as an implementation of OpenMI 1.4 and has been upgraded to FluidEarth 2 to meet the specification of this new OpenMI standard. The SDK is called the 'FluidEarth 2 SDK' and the GUI, 'Pipistrelle' (a follow up to the OpenMI 1.4 version of Pipistrelle and the OpenMI 1.4 Configuration Editor). They are Open Source and available on SourceForge (3). They are the only such open source tools available so in this sense they act as the reference SDK and GUI for OpenMI 2.0 with Windows .Net. The FluidEarth 2 project has also provided a comprehensive training website and examples, both ready for use and to act as templates for the user's own components (4).

MODELLING

FluidEarth community interaction takes place on the FluidEarth portal (5). All are welcome to join the community and contribute.

Test Cases

We now introduce some of the concepts involved in integrated modelling by looking at a set of test cases. The models have been derived to explore and illustrate different aspects without being too complicated in themselves.

The first test case is the 'Simple Pond'. It is taken from the FluidEarth training website (4) and is one of the simplest examples of an OpenMI component. The objective of this example is to allow the user to grasp some of the basic concepts of using OpenMI 2.0 with FluidEarth 2. The Simple Pond component has three arguments: capacity, current-level and flow. Water drains out of the pond as the composition proceeds through time-steps. The first composition given in the training, with the Simple Pond model given in C# and FORTRAN, shows the pond as the only component in a stand-alone composition.

A simple corollary proceeds with a composition of one Simple Pond draining into its twin. This second composition draws the user into a concept common to model coupling technologies – that of having to adapt the outputs of one numerical model before it can be connected to a second model. Since components in typical OpenMI compositions will have been developed independently and will have been designed to meet different requirements it is highly unlikely that it will be possible to simply connect the components together. Some sort of adaptation will be required in order to pass data between them. This can occur for a variety of reasons ranging from differing spatial structures to differing definitions of environmental phenomenon. In the Simple Pond example, the most straightforward of these is represented – that of a unit conversion. Pond #1 drains in centilitres, yet Pond #2 is expecting millilitres. Figure 1 illustrates this composition. OpenMI version 2.0 allows for this adaptor concept with the adaptors independent of the components.

The second test case, again taken from the FluidEarth training website, is the 'Two-dimensional Pond'. The theme is continued but a geospatial structure is added to the components. Each pond in this example offers an output array at each boundary, evenly spread across each length to represent water transfer across the entire length of each pond edge. When two such pond components are joined in a composition the action is similar – fluid will flow from one part of the pond to another as it drains into a second, identical pond component along a boundary. The nodes of the eastern boundary of the first pond match to the nodes on the western boundary of the second pond one-to-one, with values passed directly between the two.

Quillon Harpham **is a Principal Scientist at HR Wallingford specialising in Hydroinformatics. He has an MSc in Numerical Analysis (Brunel University), a BSc (Hons) in Mathematics (University of Bath) and twenty years' experience of water industry information systems.**

Paul Cleverley **is a software developer at HR Wallingford specialising in application software analysis and design. He has a BA in English and Theatre Arts and thirty years' experience in developing software applications in a broad range of industries including language development, eCommerce and hydrology.**

Removing the assumption that the arrays along each boundary edge are the same size brings in the need for an adaptor. In figure 2, the 'ten-node' eastern boundary of Pond #1 needs to be connected to the 'five-node' western boundary of Pond #2. Without a one-to-one mapping of outputs to inputs an interpolation adaptor allows values to be successfully passed between these two pond components.

The third FluidEarth 2 test case represents an example more typical of 'real world' usage of the FluidEarth 2 toolset. The OTT2D model is a two-dimensional shallow water solver employing a collocated finite volume scheme. It is used in conjunction with a sediment continuity equation

Figure 1: The simple pond composition

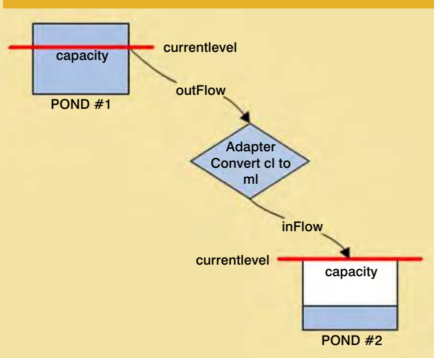
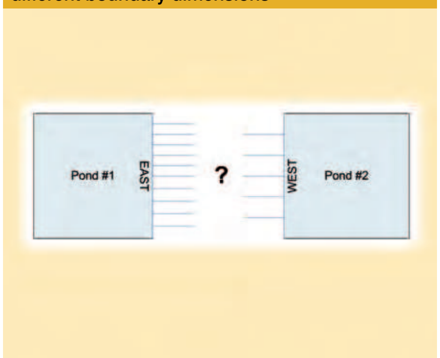


Figure 2: Connecting two-dimensional ponds of different boundary dimensions

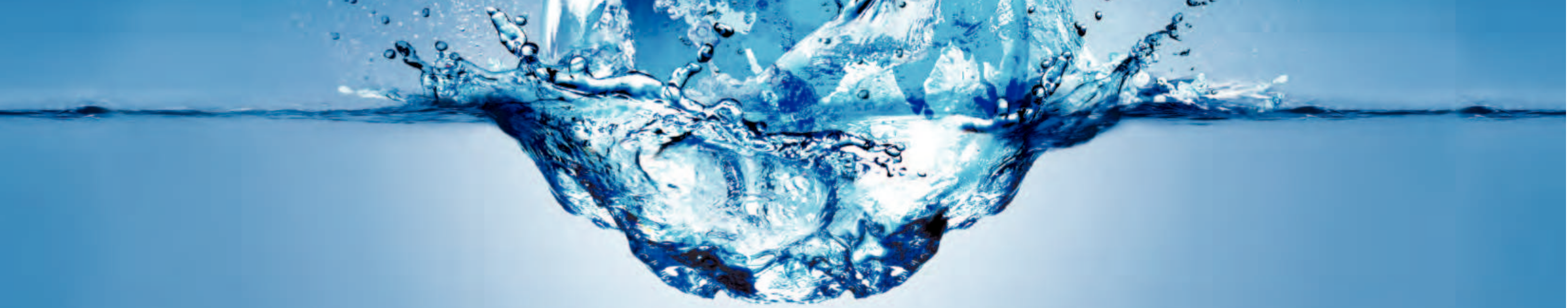


Prince Sultan Bin Abdulaziz
International Prize for Water

Recognizing Innovation

Invitation for Nominations 6th Award (2014)

Nominations open online
until 31 December 2013



www.psipw.org e-mail: info@psipw.org

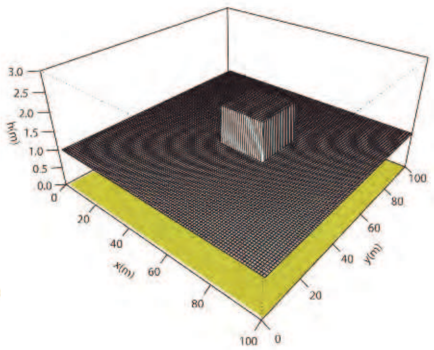


Figure 3: Bed deformation test case initial conditions

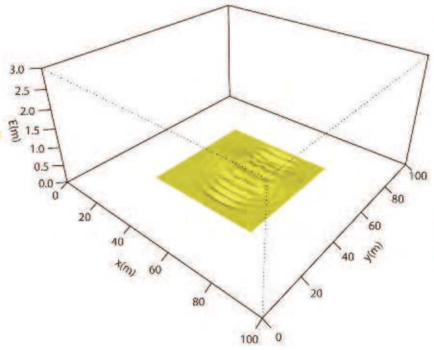


Figure 4: Total bed evolution at t=4s

solver called Exner which considers sediment movement to calculate changes in river and sea beds caused by the movement of the water. A notional cube of water is held above water at a constant depth, to create a wet-wet dam-break type scenario (see figure 3).

The dam (cube) walls are then assumed to vanish instantaneously creating a shock wave, or bore, that propagates outwards towards the domain boundaries. This causes a deformation of the bed. The meshes for the two solvers are not coincident and so the OTT2D FluidEarth 2 OpenMI component is connected to the Exner FluidEarth 2 OpenMI component via an adaptor, a bivariate interpolator. Figure 4 shows the total bed evolution after four seconds of the simulation.

For the final test case we consider a composition involving a two-way connection between components. In such two-way compositions, components pass data to each other on demand as the composition runs, with each model advancing its internal time. Component A requests data from Component B which runs through sufficient internal time-steps until it can fulfil Component A's request. Similarly Component B may reach a point where it needs to request data from Component A. Component A then runs through sufficient time-steps until it, in turn, can fulfil Component B's request. One component will be the prime driver of this composition (connected to the run trigger) and its completion will signal the completion of the composition itself.

Such a bi-directional exchange of data between components may result in deadlock: Component A is waiting for Component B to fulfil its request for data, but Component B cannot do so until it receives data from Component A. Neither component can proceed and the composition fails to complete successfully. Pipistrelle provides a solution to prevent such deadlock situations occurring: if a component is asked for information that it cannot provide by computation (for example because it would be relying on data supplied from the requesting component) then the component is forced to provide a value, even if it has to approximate.

We explore this concept with FluidEarth 2 through a composition connecting two reservoirs, A and B, by two independent channels. One channel allows water to be pumped from A to B only, the other from B to A only. The starting level of reservoir A is higher than that of B and the system is set up to attempt to reach equilibrium. Figure 5 shows a screenshot of the composition in the Pipistrelle GUI.

When the composition is run we find that water is pumped from A to B until the levels are approximately equal, yet at the point where equilibrium is expected, the system oscillates indefinitely with water levels crossing over. Figure 6 shows these results in a chart of water level against model time steps. This is an example of numerical model instability which can occur in such circumstances, often observed at transition points where water flow changes direction.

Figure 5: Two-way reservoir coupling in Pipistrelle

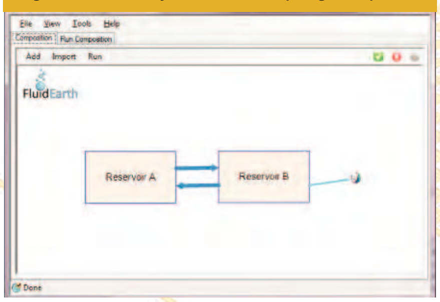
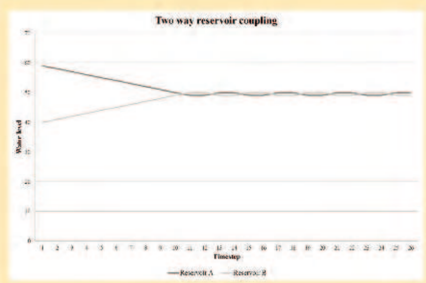


Figure 6: Two-way reservoir coupling composition results



Summary and Acknowledgements

FluidEarth 2 is an implementation of OpenMI version 2.0 which seeks openness, flexibility and usability. The examples using the FluidEarth 2 SDK and Pipistrelle given above, range from simple one-way compositions to those more typical of real industry or academic requirements. They have been built in C# and FORTRAN (with Visual Basic usage seen as a corollary) where the model coupling process has been improved and made accessible to less technical users. Using the Pipistrelle GUI, compositions can be built utilising compatible components from different suppliers in a high usability environment. The training website includes a detailed level of instruction, especially for use of the SDK since this is the most involved procedure, tending to be the most esoteric.

This article has been summarised from a paper: Harpham, Q.K., Cleverley, P., Kelly, D. (2013) 'The FluidEarth 2 Implementation of OpenMI 2.0', currently under submission to the Journal of Hydroinformatics.

The FluidEarth 2 toolkit (Pipistrelle and the FluidEarth SDK) are open source developments freely available on SourceForge (3). The code was developed for HR Wallingford by Adrian Harper of Innovyze.

FluidEarth 2 was co-funded by the EC 7th Framework Programme DRIHM Project, Grant Number 283568.

OpenMI is governed by the OpenMI Association and OpenMI 2.0 is currently in the final stages of becoming a standard ratified by the Open Geospatial Consortium (OGC).

References

OpenMI Association (2012), Short History, OpenMI Association Website, http://www.openmi.org/new-to-openmi#TOC-Short-history. Accessed 21st August 2012.

OpenMI Association (2010), OpenMI SourceForge Project, http://sourceforge.net/projects/openmi/. Accessed 18th October 2013.

HR Wallingford (2012), FluidEarth SourceForge Project Open Source Development for FluidEarth, http://sourceforge.net/projects/fluidearth/. Accessed 18th October 2013.

HR Wallingford (2012), FluidEarth eLearning Website, http://eLearning.fluidearth.net, Accessed 25th October 2013.

HR Wallingford (2008), FluidEarth Portal, http://fluidearth.net, Accessed 25th October 2013.