

The OpenMI 2.0 Standard for Integrating Numerical Models

Harpham, Q.K (HR Wallingford); Hughes, A. (BGS) and Moore R.V.

Abstract

The purpose of this paper is to introduce, explain and promote the Open Modelling Interface (OpenMI) version 2.0 standard for coupling environmental numerical models (simulations of environmental processes). It is intended to be accessible to readers of all levels of experience. During recent decades it has been recognised that the environment is made up of a complex set of interconnected processes. Therefore, understanding it requires not only understanding of the processes in isolation, but also the interactions between these processes. Traditional methods of simulating such environmental interactions have included passing the outputs of one numerical model into another or creating a single 'super-model' covering a variety of processes. OpenMI provides a standard method which could be applied to independent numerical model components and allow them to exchange data so that they could interact and influence one another. This is achieved without fundamental changes to the core of the components themselves.

Keywords

Open Modelling Interface

OpenMI

Integrated modelling

Model linking

Model coupling

Decision support systems

Open standards

Open source

Open Geospatial Consortium

Software Availability

The paper definition of OpenMI can be obtained from its associated OGC (Open Geospatial Consortium) standard definition at <https://www.opengeospatial.org/standards/openmi>.

Object interface libraries for OpenMI 2.0 standard can be obtained from its associated Source Forge project, <https://sourceforge.net/projects/openmi/>. They have been prepared in Java and C#, together with other supporting material.

The OpenMI Association website is available at <http://openmi.org>.

60
61
62 HR Wallingford's FluidEarth implementation of OpenMI can be obtained from its associated Source
63 Forge project, <https://sourceforge.net/projects/fluidearth/>. A comprehensive training website with
64 example numerical model compositions is available at <http://eLearning.fluidearth.net>.
65
66

67 68 Introduction

69 Purpose

70
71 The purpose of this paper is to introduce, explain and promote the appropriate use of the Open
72 Modelling Interface (OpenMI) version 2.0 standard for coupling numerical models. It is intended to
73 be accessible to readers of all levels of experience, from those writing and implementing numerical
74 models through to those exploring the new discipline of integrated modelling for the first time. The
75 paper supplements the full specification of the standard which was developed and is maintained by
76 the OpenMI Association (OA) and is published by the Open Geospatial Consortium (OGC) (OGC
77 OpenMI, 2013) as an OGC standard. It is an open standard and is available through the OGC website.
78
79

80 The OpenMI standard was developed in order to facilitate the understanding of interacting
81 environmental processes. While it was conceived within the domains of hydraulics and hydrology, it
82 was appreciated from the outset that the need was for a mechanism that could link across domain
83 or disciplinary boundaries. Indeed, the OpenMI user community already extends well beyond the
84 original domains and, indeed, beyond those concerned with simulating the natural environment.
85 However, it was the growing need to understand the interconnectedness of environmental events
86 and processes which prompted its inception.
87

88 OpenMI 2.0 in Context

89 Traditionally, environmental simulation models have been created by scientists working in
90 independent specialist groups. Today, they nearly all take the form of computer programs and are
91 referred to collectively as numerical models. These models usually consist of an algorithm that
92 simulates one or a few specific processes articulated in computer code. The program receives the
93 description of the situation it is to simulate from input data files and/or a user interface. Results
94 from the simulation are usually returned via one or more output data files.
95
96

97 During recent decades it has been recognised that the environment is made up of a complex set of
98 interconnected processes (ref Moore, Harpham etc.). Changes to one environmental phenomenon
99 at one location can have a fundamental effect on other phenomena at another location (Meiburg
100 2007). As a result, environmental science is being challenged to address increasingly complex
101 questions, such as "will the predicted rise in temperature due to climate change lead to Malaria
102 being a health problem in more countries?" (Moore and Hughes, 2017). The traditional method of
103 simulating such environmental interactions has been to run the process models separately, one
104 after the other, the results of the first becoming the input of the second. (see Anastas, 2010 and also
105 Voinov, 2010). For example, in order to simulate a flooding scenario, one numerical model
106 simulating rainfall passes its output into another numerical model which understands river
107 catchment drainage using rainfall as its input. A chain of models is constructed with the output of
108 the 'upstream' model simply requiring interpolation or re-formatting in order for it to be made
109 compatible with its 'downstream' counterpart.
110
111

112 By combining different models together the end result can answer complex questions more
113 effectively than by using a single existing model. Further, the solution of complex questions require
114 holistic solutions drawn from different disciplines (Laniak et al., 2013). Therefore, it may be
115
116
117
118

119
120
121 preferred to combine numerical models together into one, larger, more comprehensive
122 implementation. So instead of passing the rainfall between the model producing rainfall and the
123 model receiving the rainfall data, the computer codes are combined into a single numerical model
124 which performs both modelling functions. Internally, the same function will be being performed, but
125 the passing of the rainfall will occur 'in memory' as the combined models execute. This second
126 option opens up the possibility of allowing the processes to influence one another as they proceed
127 through time. It may be that the phenomena being described are closely interlinked and having a
128 two-way impact on each other. For example, if a torrent of water rushes against the side of the
129 building then the presence of the building will have an effect on the passage of the water. The water
130 may cause a wall of the building to collapse and so the presence of the water will affect the building.
131 To correctly simulate this, the numerical model describing the state of the building structure will
132 need to be combined with that describing the flow of the water. However, it is likely that the
133 numerical model describing the flow of the water will come from one community of experts and the
134 model describing the building structure will come from another. There may be no appetite to
135 combine these into a single 'super-model'. Further, such a combined model will be exponentially
136 more difficult to maintain and the freedom of the modelers to innovate will be severely inhibited.
137 Thus, if these two model codes could be kept separate yet enabled to exchange data as they ran in
138 parallel, behaving as a single composition, then each community could develop and innovate their
139 own model without reference to the other.
140
141
142

143 It was recognised that considerable investment had been made into developing many numerical
144 models by the organisations responsible for them. This investment couldn't simply be discarded in
145 favour of developing a new generation of environmental numerical models from scratch. Therefore
146 a solution which enabled model integration was sought which could be applied to any numerical
147 model, irrespective of technology, history or ownership – a solution which respected intellectual
148 property and diversity.
149

150 The Case for Standardisation

151
152 Users have required something more capable than manual data transfer between models and
153 something more manageable and sustainable than 'super-models'. To this end, some numerical
154 model developers have joined two or more established models together, keeping the core of each
155 broadly separate but operating them together. For example, Sophocleous et al. (1999) demonstrate
156 a combined MODFLOW-SWAT model combination where MODFLOW simulates groundwater and
157 SWAT surface water. Other practitioners use web processing services to pass data between linked
158 model components via the internet as illustrated by Nativi et al. (2013). The original intention behind
159 OpenMI was to provide a standard method which could be applied to independent numerical model
160 components and allow them to exchange data whilst they are running. This was to be achieved
161 without fundamental changes to the core of the components themselves thus still allowing them to
162 be run independently, when required.
163

164
165 This situation allows experts to remain within their domains of expertise, each continually innovating
166 to produce successive generations of numerical model programs which increasingly accurately
167 describe the phenomena they represent. Very often, such changes do not alter the data types taken
168 in or output by the model. The changes therefore make no demands on the authors of other linkable
169 models. Achieving this successfully depends on the interface between the models being tightly
170 defined and that definition being readily accessible by all. Such common understanding drives the
171 need for standardisation, thereby allowing such compositions to be built and executed correctly.
172 Thus, each set of experts is required to engineer their software according to a specified standard.
173 This standard would have to contain all the elements necessary for the inter-model communication,
174
175
176
177

178
179
180 yet allow each set of domain experts the freedom to apply their own numerical implementation of
181 the associated mathematical theories, as well as the necessary functional elements to achieve an
182 accurate simulation.
183

184 It could be suggested that asking different domain experts to adopt a common standard for software
185 engineering would hamper creativity. If they are restricted in how they can implement their
186 numerical models then surely that restriction would hamper further experimentation and discovery?
187 The authors submit that the opposite would be the case. Consider the example of music and musical
188 notation. The decision to limit musicians from a continuous distribution of an infinite number of
189 tones to just twelve notes – the seven natural notes plus five sharps and flats, notwithstanding the
190 implementation of this across multiple octaves – serves to enable creativity. Having described the
191 twelve notes, musicians then discovered that they could be combined to articulate chords as well as
192 describing patterns and scales. They then created the common language of musical notation to
193 communicate this. These processes enabled creativity rather than hampering it. It would seem
194 absurd now to abandon this and revert entirely to music with no fixed notes or notation. Such is also
195 the case for numerical modelling.
196
197

198 The History of OpenMI

199 The first published OpenMI standard was version 1.4 (Gregersen et al. 2007). It was the result of the
200 European Commission's Framework 5 HarmonIT research project. As well as proving the concept of
201 model integration through a common standard, HarmonIT also prototyped many tools to facilitate
202 making existing models into 'linkable components' and then building and running linked models or
203 'compositions'. The success of OpenMI 1.4 prompted a second European project, this time part of
204 the LIFE programme, OpenMI-LIFE. OpenMI-LIFE was given a 'best of LIFE programme' award in
205 2012. The main thrust of OpenMI-LIFE was to demonstrate the OpenMI standard being used in a
206 real-world environment on real-world problems. However, a lot of thought was given to how the
207 development of OpenMI would continue once OpenMI left the research environment and had to
208 stand on its own feet. How could its development be continued? How would its users be supported?
209 How could the outside world be convinced that these mechanisms were in place, working and would
210 be sustained? At that time adoption of the OpenMI represented a significant commitment of
211 resources. To address these points the OpenMI Association (OA) was founded as a legal entity under
212 Dutch law and took ownership of the IPR of OpenMI. To demonstrate that it was a functioning
213 organisation, its first task within the LIFE project was to produce version 2.0 of OpenMI.
214
215
216

217 As that moved towards completion, the Open Geospatial Consortium approached the OA, explained
218 its growing awareness of the importance of the time dimension and of modelling to its work and
219 suggested that the OA and OGC collaborate. The OA was delighted as this would enable it to publish
220 the OpenMI under the banner of an internationally recognised standards organisation and hence
221 reach a world-wide audience spanning almost every discipline and both the public and private
222 sectors. This has now been achieved.
223

224 In parallel, a number of influential hydraulic and hydrological institutions and laboratories including
225 Deltares (<https://www.deltares.nl/en/>), DHI (<https://www.dhigroup.com/>) and HR Wallingford
226 (<http://www.hrwallingford.com/>) created implementations of the standard and applied it to their
227 own numerical model inventories. Since its release, OpenMI 2.0 has been downloaded over 13,000
228 times, with active interest from over 30 countries. Interest has been observed from a wide range of
229 specialisations ranging from internal combustion engines to human biology and the number of
230 implementations increases.
231
232
233
234
235
236

Overview of OpenMI 2.0

Standards can only enable creativity and interoperability if they are properly defined and kept as simple as possible. If this is not the case then the overhead of using them can stifle any advantage (see Iwanaga (2018)). Each aspect of the standardisation must be necessary and proportionate. OpenMI therefore incorporates aspects necessary and proportionate for independent numerical models to be integrated.

At its heart, integrated modelling is about converting a question into a workflow and providing a quantified solution. This involves combining data and models such that the solution can be obtained. The model can be thought of as a unit operation (Figure 1) and be seen as having two facets: model codes which represent system processes / behaviour programmatically and model instances which are the application of the model code to a particular area or problem. By considering a model as a unit operation and defining the input and output data such that the output from one model code can be passed to another, then a linked series of models and an associated workflow can be created. OpenMI uses this paradigm by observing that a composition of models can be thought of as a unit operation made up of independent components.

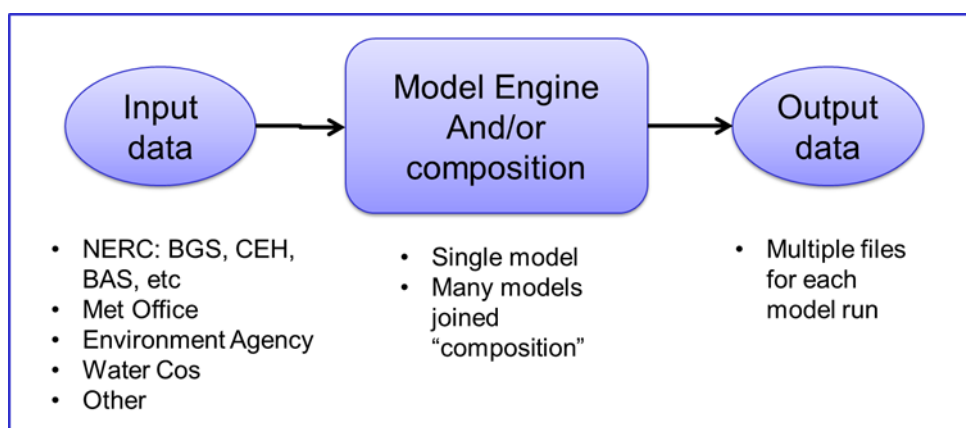


Figure 1: Model as a "unit operation"

OpenMI 2.0 allows those writing numerical models to adapt them so that they can be combined into compositions containing other models that have been similarly modified. In technical terms OpenMI is a set of object interfaces that will form a wrapper around each numerical model code. It can be implemented in any object oriented programming language. When made OpenMI 2.0 compliant, a numerical model becomes a 'Linkable Component' which can then be run in compositions including other Linkable Components. Some of the key ideas and principles implemented by OpenMI 2.0 in forming Linkable Components are given in the sub-sections following.

Control and Structure

The OpenMI 2.0 interfaces act as a code wrapper around the core numerical model code. Control of the Linkable Component's ability to perform typical functions such as Initialise, Run, Finalise (IRF) are offered to an external, unspecified entity. Usually, numerical models are supplied with an interface, often graphical (a GUI) to enable the user to set parameter values (such as the location of input datasets), set the model running and monitor its progress. However, if the model is to run in an OpenMI 2.0 composition, even if an instance of it has to be created using a GUI, it must be possible for an external entity to control the process of executing the model code.

Achieving this requires that the model code has a certain characterisation:

- Parameters for the model to run, which are not being offered to other components – such as the location of input or output files – must be independently accessible once the model is set up to run.
- The 'initialise' and 'finalise' functions need to be distinct and control of them available to the external entity which will be running the code.
- The core model function (often consisting of calculating values at a sequence of intervals in time) must also be distinct and control of it available to the external entity.

Indeed, if an OpenMI-compliant component is to run in one of the existing GUIs then it shall be associated with an XML file (called an 'OMI' file) which can be successfully validated against the LinkableComponent.xsd schema (Annex B, OGC OpenMI, 2014).

Exchange Items

Numerical models will produce a set of results as a set of values for a defined phenomenon or parameter, for example air temperature, water depth, current direction or shoreline position. In some cases, this can be a very long list. As part of making a numerical model OpenMI compliant, the model developer chooses which of their output variables – belonging to the Linkable Component – to make available to other numerical models. Each one of these becomes an 'Output Exchange Item', see IBaseExchangeItem and IBaseOutput (OGC OpenMI, 2014, section 6.9).

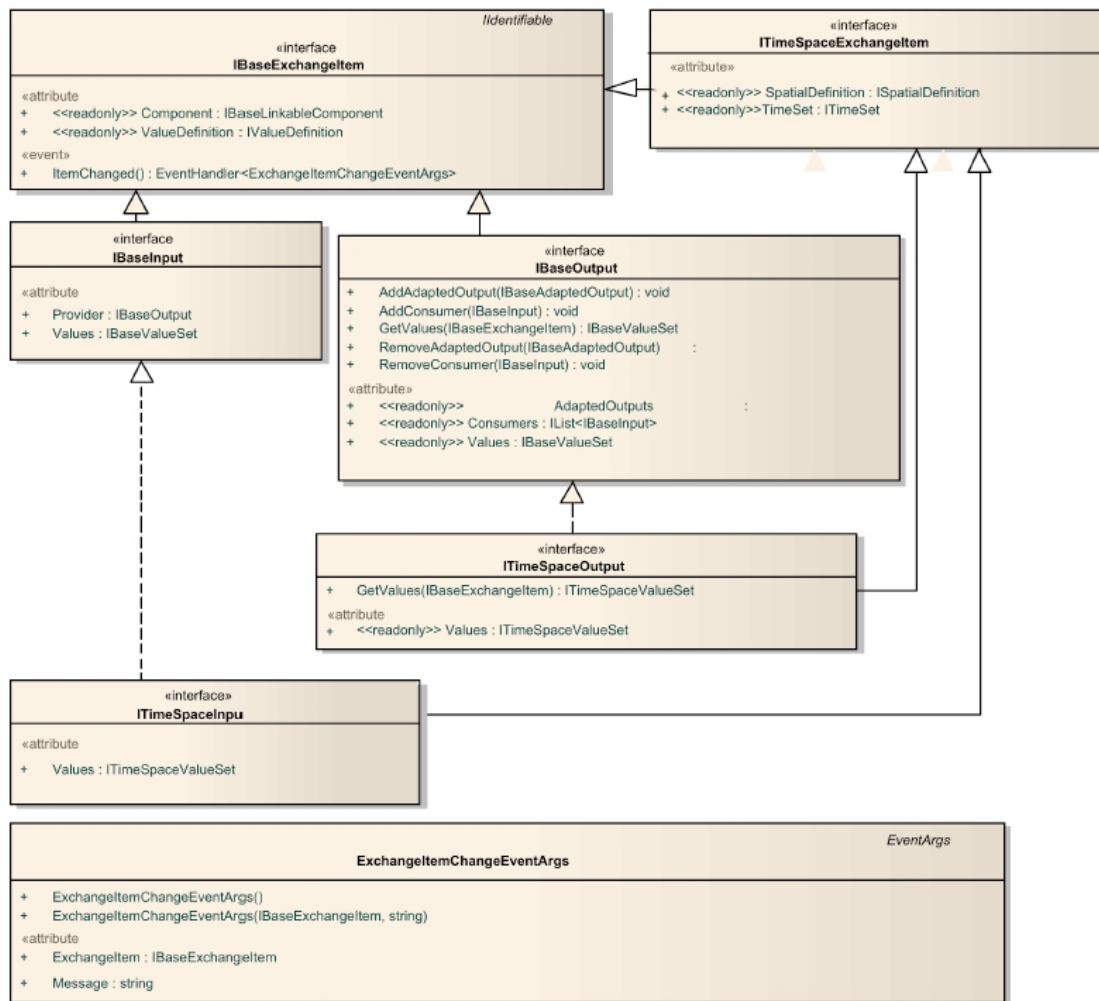


Figure 2: UML Class Diagram for IBaseExchangeItem

Similarly, numerical models are able to – and usually require – phenomena or parameter values from external sources, for example a simulation of near shore waves will require a bathymetry dataset describing the sea bed. This phenomenon may be variable in time and it may be possible for a numerical model to receive this data from a second numerical model. When a model is made OpenMI compliant, such inputs are designated ‘Input Exchange Items’, see IBaseExchangeItem and IBaseInput (OGC OpenMI, 2014, section 6.9).

Only Exchange Items will be able to be passed from one numerical model to another via the OpenMI interface.

Parameters and Units

If two model components are exchanging and using values associated with a phenomenon across an Exchange Item, then the phenomenon itself and its associated units must be clearly and mutually defined. We cannot have a situation where one numerical model is passing ‘Wave Height’ to another model where the former understands it to be ‘Significant Wave Height’, measured in metres and the latter understands it to be ‘Maximum Wave Height’, measured in feet. Thus, OpenMI breaks down the phenomena passed between the exchange items into its DimensionBase as part of the IValueDefinition interface (Length in metres, Mass in kilograms, Time in seconds, ElectricCurrent in amperes, Temperature in kelvin, AmountOfSubstance in moles, LuminousIntensity in candela, Currency with no base quantity) (Table 3, OGC OpenMI, 2014). The IQuality interface inherits from the IValueDefinition interface to allow non qualitative values (e.g. high, low) to be passed, for example, as strings. In the case where the units are dimensionless, represent logarithmic scales or are otherwise problematic when expressed in SI, then extra attention should be paid to the descriptive part of the unit to clarify understanding of the meaning of its value.

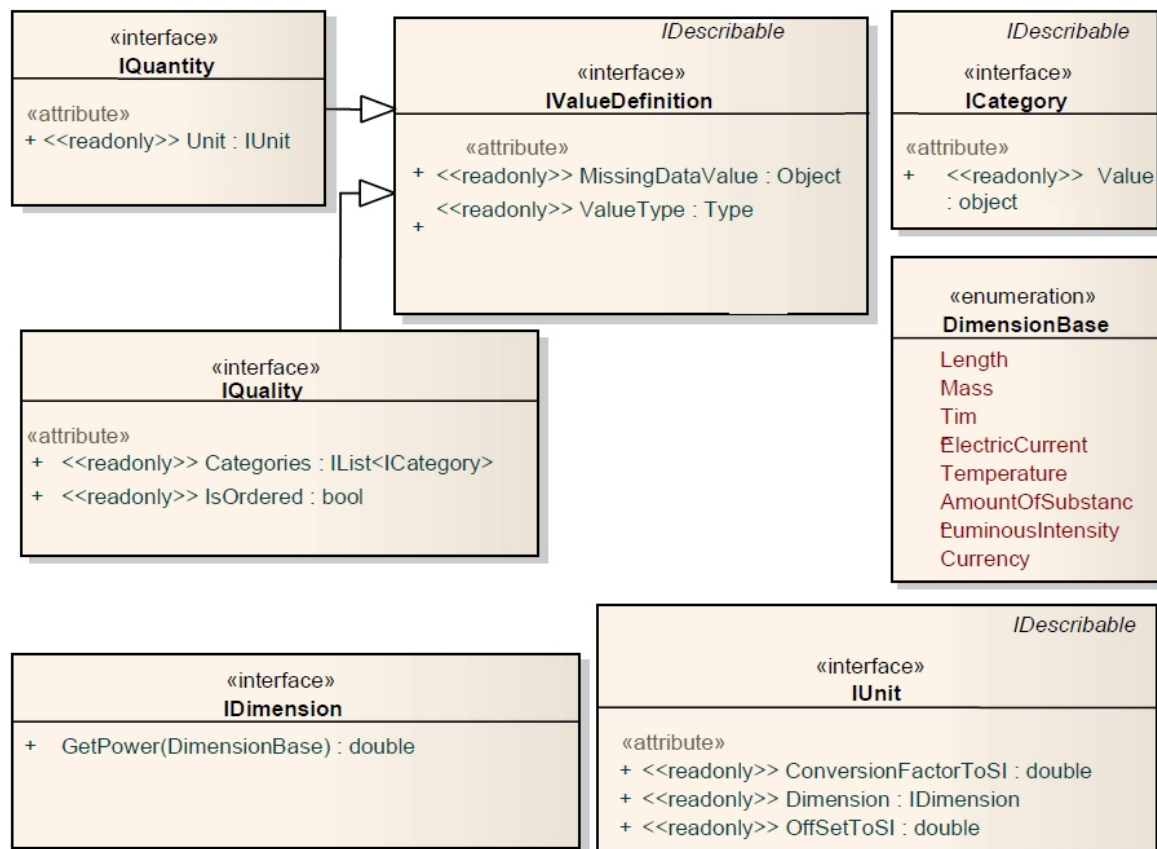


Figure 3: UML Class Diagram for IValueDefinition

This, nor indeed any, definition can be perfect and completely prevent errors in definition or unit, but this structure is intended to offer enough rigour to maximise the opportunity to trap and correct such inconsistencies.

Spatio-Temporal Structure

Numerical models typically produce their results according to a specified (or assumed) spatio-temporal structure. Harpham (2018) defines a vocabulary for describing these structures as given in Table 1. Such vocabularies constitute important information which can be referenced in metadata associated with models (Harpham and Danovaro, 2015).

Spatial Structure	Point	PointSet	Polyline	PolylineSet	Polygon	PolygonSet (Grid, Mesh)
Temporal Variation	Point Series	PointSet Series	Polyline Series	PolylineSet Series	Polygon Series	PolygonSetSeries (GridSeries, MeshSeries)
Temporal and Spatial Variation	Point Track	PointSetTrack / Developing PointSet	Polyline Track	PolylineSet Track / Developing PolylineSet	Polygon Track	PolygonSetTrack / Developing PolygonSet (Adaptive Grid, Adaptive Mesh)

Table 1: Controlled Vocabulary for Spatio-Temporal Data Structures

The most common of these are well understood and established such as:

- PointSeries, describing a time series of values associated with a single, unmoving point in space, such as readings from a rain gauge;
- GridSeries (given as a special case of a PolygonSetSeries), describing timeseries output associated with points arranged in a rectangular grid with one value for each point for each timestep;
- PointTrack representing a point moving with time, such as the position of a moving vessel.

However, more complex spatio-temporal structures are produced by some numerical models. For example, a probabilistic shoreline erosion model would produce output at a series of timesteps. Output at each timestep is a set of polylines describing the possible position of the shoreline at different rates of erosion. Thus the output across all timesteps could be described as a 'PolylineSetTrack' or 'DevelopingPolylineSet', as illustrated in Figure 4.

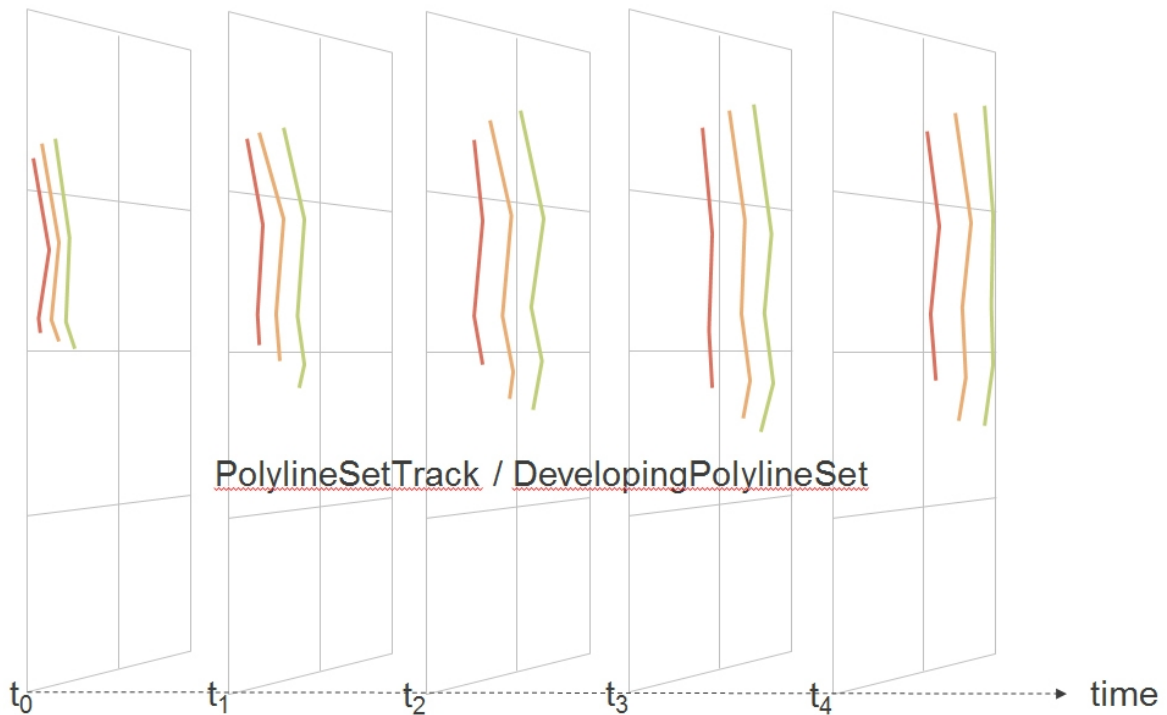


Figure 4: PolylineSetTrack – Probabilistic Progressive Shoreline Position

OpenMI is designed to maximise flexibility and so, to incorporate all common (and less common) numerical model outputs in one, two and three dimensions, the spatial and temporal components have been separated. Indeed, the core OpenMI standard does not assume that the numerical model will have a timestepping aspect at all; this is offered as one of the optional extensions to the standard – albeit a commonly used extension.

OpenMI 2.0 breaks down the spatial aspect of numerical model outputs into its basic spatial constructs. This is governed by the 'ISpatialDefinition' interface (OGC OpenMI, 2014, section 6.4) including an 'ElementCount', 'SpatialReferenceSystem' and 'Version'. The 'Element Set' of the spatial structure 'IElementSet' includes management of indexes, counts and coordinates, with each element enumerated into five types: IdBased, Point, Polyline, Polygon and Polyhedron.

Where required, time series values for the Element Set are described by the ITimeSpaceValueSet interface (OGC OpenMI, 2014, section 6.6). Also included and of particular importance for running time series compositions of numerical models is the TimeHorizon property of the ITimeSet interface (OGC OpenMI, 2014, section 6.5), which describes the full time interval governed by the numerical model.

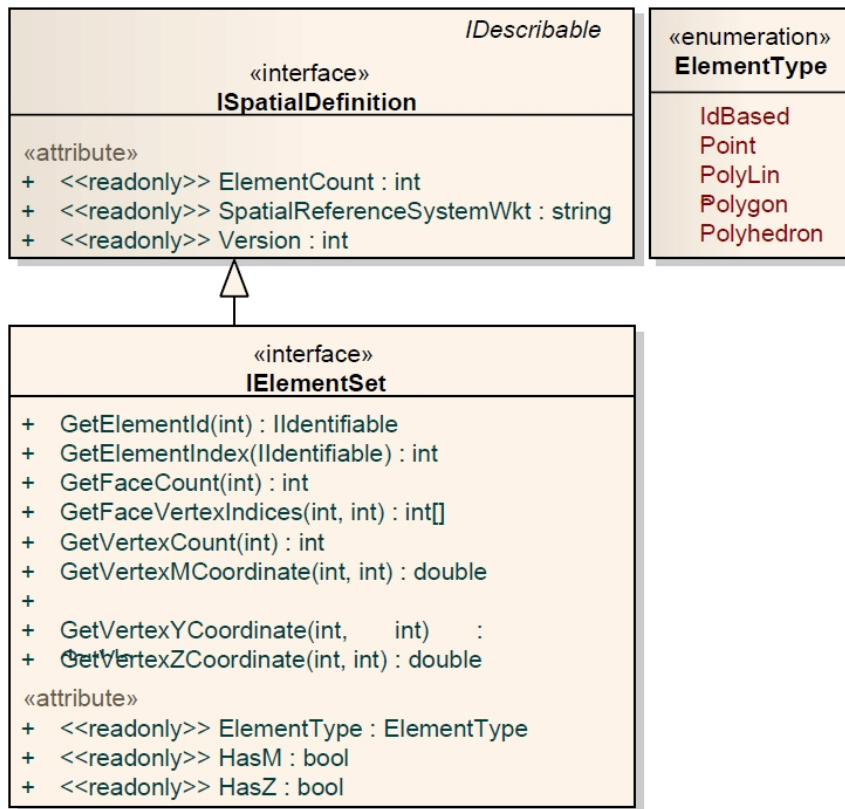


Figure 5: UML Class Diagram for ISpatialDefinition

Adaptors

Two independently developed OpenMI 2.0 compatible numerical models can have their inputs and outputs connected according to the desire of the modeller: input exchange items to output exchange items. However, this is unlikely to be feasible without adjustment. For example, the first model may produce its results as a rectangular GridSeries and the second model may wish to receive them as a PolylineSeries; the first model may produce its results in metres and the second model may wish to receive them in feet; the two models may use a different vertical datum. Therefore, to be connected correctly an 'adaptor' will be required across the connection between input and output exchange items in order to perform the necessary adjustments, see Figure 6 where the notation used is that adopted by Harpham and Hummel. Such adaptors are created using the interface IAdaptedOutputFactory and managed by the interface IBaseAdaptedOutput (OGC OpenMI, 2014, section 6.10).

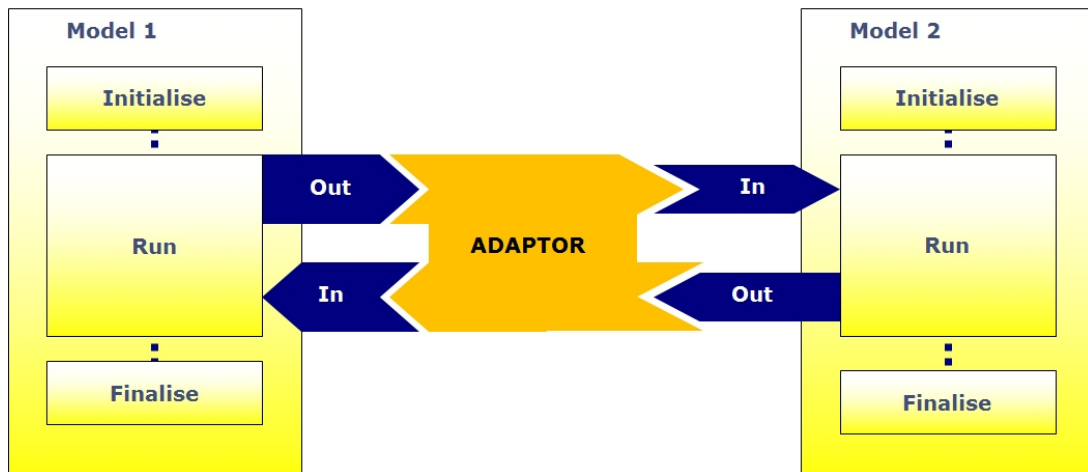


Figure 6: Use of an adaptor to handle differences between input and output exchange items, Harpham / Hummel notation.

It is possible to nest multiple adaptors and apply them in turn and in combination. For example, one adaptor may perform a unit conversion required between two models and another may perform a datum conversion additionally required by a third model. This situation is illustrated in Figure 7 using the notation adopted by Harper. It seems that only individuals with family names beginning with the letter 'H' are able to create notation for adaptors.

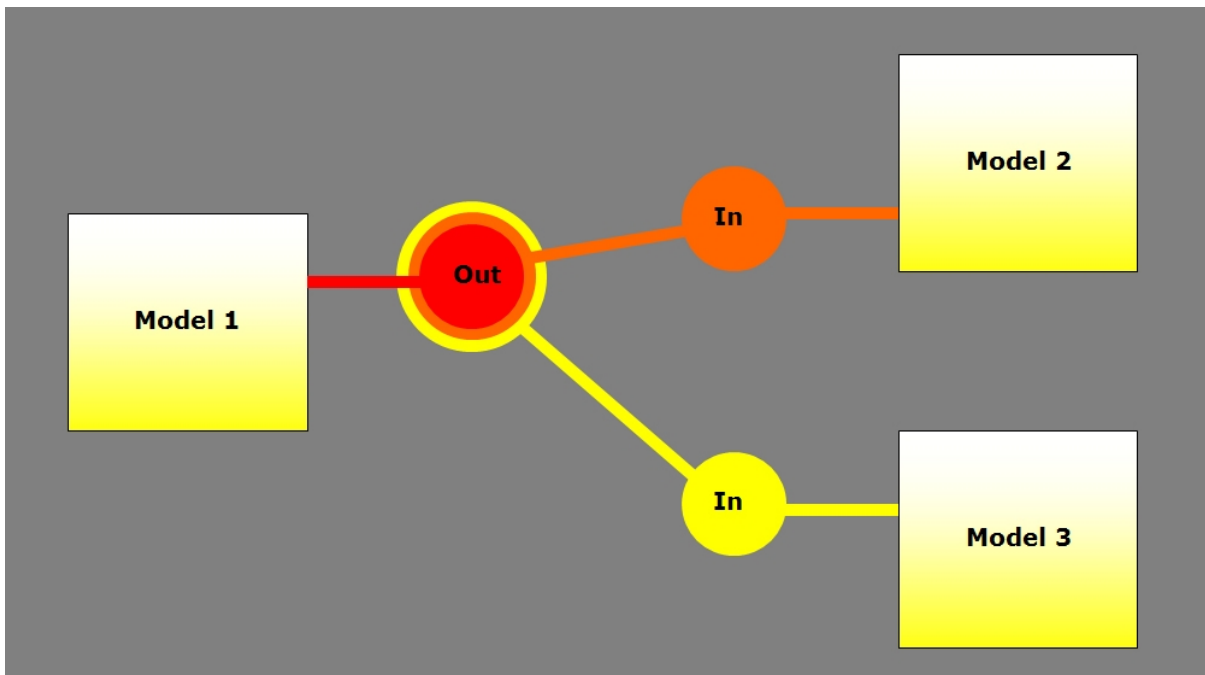


Figure 7: Nested adaptors shown using the Harper notation.

Changes from OpenMI 1.4:

There are a number of key differences from OpenMI 1.4. In addition to some lower level technical changes, the main aspects are as follows:

- Base Interfaces and Extensions – A set of minimum ‘base interfaces’ for compliance, plus the addition of extensions (including an extension covering time and space dependent components). In particular, the essential OpenMI component is no longer forced to be time and space dependent, making the standard considerably more flexible and extensible. This allows different types of components to be incorporated e.g. those which vary in time and not in space; those which vary in space, but not in time or those which vary in both time and space.
- Adaptors – Taking over from the role of ‘Data Operations’ in OpenMI 1.4, ‘Adapted Outputs’ allow multiple, distinct adaptations, separate from the components themselves and the link, to take place. Again, this makes the standard more flexible and allows outputs and adapted outputs to be re-used by more than one OpenMI component.
- Events – OpenMI 1.4 had its own in-built event system. For version 2.0 the standard .NET and Java event mechanisms are utilised.
- Data Definitions – Values passed in OpenMI 1.4 could only be quantitative. OpenMI 2.0 supports qualitative information (e.g. text values such as ‘high’ and ‘low’). It also incorporates an extended element set to follow the OpenGIS standard with no distinction between 2d and 3d spatial structures, improved handling of timestamps and timespans, and a more flexible value set to include objects as well as scalars and vectors.

Making new and existing models OpenMI 2.0 compliant

The original vision and purpose behind OpenMI was to create a method for easy integration of both new and existing numerical models – to build on the existing investment in the development of numerical models. The method provided is to make each numerical model ‘OpenMI 2.0 Compliant’, that is, make each numerical model code into an OpenMI 2.0 Linkable Component, adhering to the standard. Compositions of multiple components can then be put together and run as a collective with data exchange between linked components in the composition.

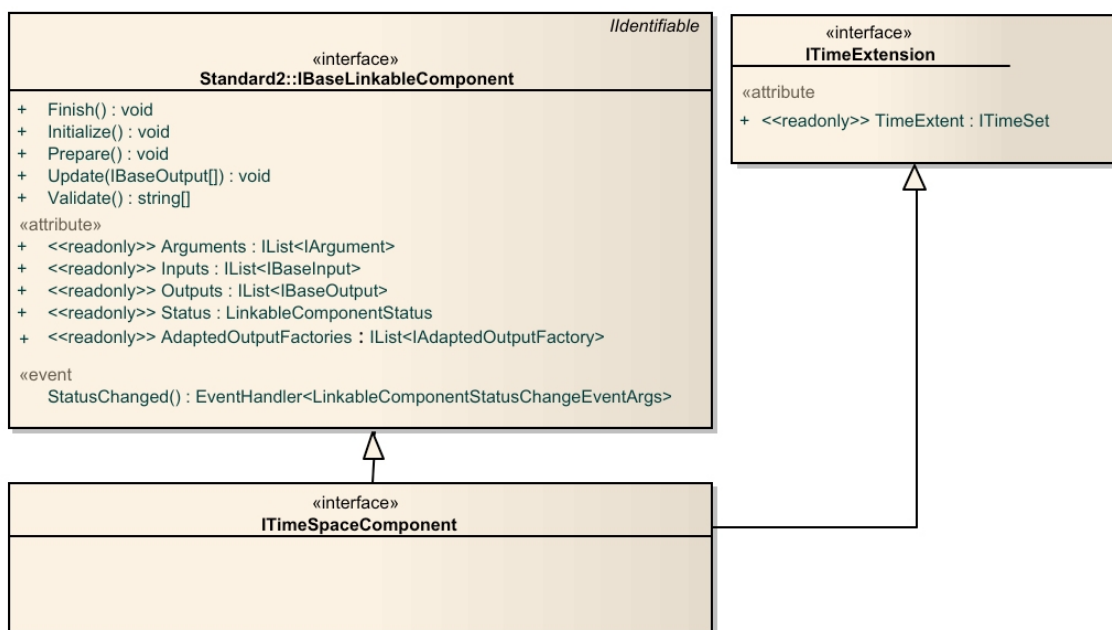


Figure 8: UML Class Diagram for IBaseLinkableComponent

709
710
711 A numerical model (or equivalent) is OpenMI 2.0 compliant if it implements a set of interfaces that
712 can connect to and interact with the OpenMI 2.0 component interface IBaseLinkableComponent or
713 its specialisations (such as the ITimeSpaceComponent). These interfaces are described in Clause 6,
714 'OpenMI Requirements Classes' of OGC OpenMI (2014). In particular, requirements for compliance
715 can be summarised as follows:
716

- 717 1. An OpenMI 2.0 compliant component shall implement the mandatory core requirements
718 given in clause 6 of OGC OpenMI (2014).
- 719 2. Two further optional *core* interfaces (IManageState and IByteStateConverter) may be
720 implemented. If implemented then this must be done according to the OGC OpenMI (2014)
721 specification.
- 722 3. Optional OpenMI 2.0 extensions are also given. If implemented, this must be done according
723 to the OGC OpenMI (2014) specification.
- 724 4. An OpenMI 2.0 Compliant component (including its extensions) shall, when compiled,
725 reference the OpenMI.Standard2*.DlIs (.Net framework 2.0 or higher) or OpenMI-
726 standard2*.jars (java 1.5 or higher) as released by the OpenMI Association (OpenMI Source
727 Forge).
728
729

730 Thus, making a numerical model OpenMI 2.0 compliant can be achieved in two steps:
731

- 732 • **Step 1:** Prepare the numerical model code to be consistent with the principles outlined in
733 the above and for mapping to the OpenMI 2.0 interfaces i.e. with respect to Control and
734 Structure, tidy and accessible arrangement of input and outputs to be accessed through
735 Exchange Items, clear definitions of Parameters and Units, clear definition of Spatio-
736 Temporal Structures, and understanding of any potential requirements for use of Adaptors.
737
- 738 • **Step 2:** Connect the relevant attributes of the numerical model into the OpenMI 2.0
739 interfaces i.e. 'wrap' the model. Software Development Kits (SDKs) are provided with
740 implementations of OpenMI to aid developers in this task (Harper et al., 2016; Harpham et
741 al., 2014).
742

743 If the numerical model is being written from scratch then the principles required by Step 1 can be
744 employed directly and the model written according to the OpenMI 2.0 standard; if the numerical
745 model already exists then these principles will need to be retrofitted.
746

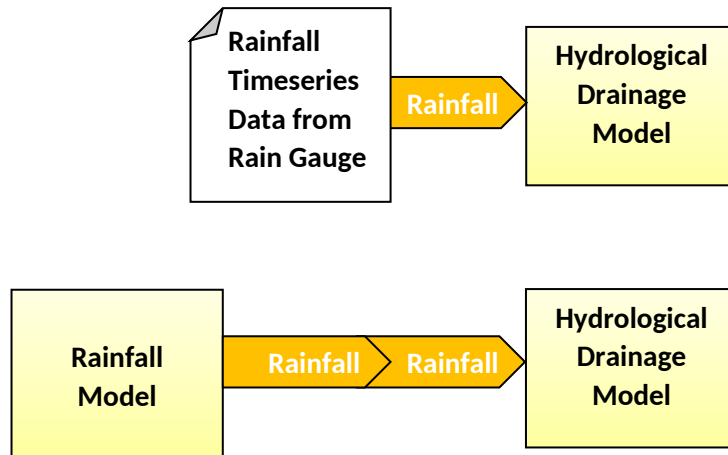
747 Creating and Running OpenMI 2.0 Compositions

748

749 OpenMI 2.0 components can be run individually, as single numerical models, taking input from the
750 set of supporting data files as standard for the model or they can be run as compositions of multiple
751 components which are able to pass data between them as they run, across Exchange Items. It is the
752 responsibility of the modeller to ensure that any connections are scientifically valid. Although
753 OpenMI includes low-level definitions of each aspect of this connection (e.g. spatial structure,
754 parameter), it can provide no specific validation. Indeed, other more subtle aspects, such as how the
755 data is represented within each spatial element, must also be considered when connections are
756 constructed. For example, if the spatial element is a polygon – say a grid cell – then is the value
757 taken by the cell attributed to its centroid or its corners? If the corners all have different values, then
758 which of these is passed to another model which has a single point linked to the grid cell?
759

760 When an OpenMI component (numerical model) takes input from another component then this
761 overrides the associated input it would otherwise get from its supporting data (configuration or
762 input data file), as illustrated in Figure 9. The Hydrological Drainage Model would usually be set up
763 taking driving data from a rainfall time series, perhaps from an in-situ instrument. This input can be
764
765
766
767

768
 769
 770 validly replaced by input from another OpenMI 2.0 component which is a numerical model
 771 producing rainfall time series output. Thus, the rainfall Input Exchange Item from the Hydrological
 772 Drainage Model is connected to the rainfall Output Exchange Item from the Rainfall Model. The
 773 Hydrological Drainage Model will then produce its output in its usual way.
 774
 775



791 Figure 9: Replacing file-based inputs with a connection across Exchange Items.
 792
 793

794 Implementations of OpenMI 2.0

795 A number of implementations of OpenMI 2.0 are available. These usually consist of i) a Software
 796 Development Kit (SDK) to facilitate the wrapping of numerical models into OpenMI 2.0 components
 797 and ii) a Graphical User Interface (GUI) to assemble compositions of components into compositions
 798 and to run these compositions. The GUI acts as the unspecified external entity which controls the
 799 composition and the components within it.
 800
 801

802 HR Wallingford's FluidEarth (Harpham et al., 2014; Harper et al., 2016) is a typical implementation of
 803 OpenMI 2.0. It is provided with the FluidEarth SDK and a GUI called Pipistrelle, with the open source
 804 code available at <https://sourceforge.net/projects/fluidearth/>. Since the OpenMI 2.0 standard is
 805 flexible, each implementer has the freedom to apply it according to their particular needs. The
 806 FluidEarth implementation uses two sets of configuration files so that each OpenMI 2.0 component
 807 is accompanied by two files, a .omi file and a .xml file which offer configurable parameters and a
 808 means of engagement with the user interface, Pipistrelle. One of the components, usually one with a
 809 time horizon starting earlier and finishing later than all the others, is connected to a 'Trigger'. This
 810 component is used to start the composition running and will make requests from other components
 811 in the composition in an attempt to pull data from them to satisfy its own requirements. In turn,
 812 components will make requests from each other as the composition progresses until it is able to
 813 complete. This process is demonstrated by a simple composition involving two ponds (pondWest
 814 and pondEast) where pondWest is slowly draining into pondEast. The composition progresses
 815 through time steps with pondEast pulling data from pondWest. The sequence diagram for the main
 816 functions is shown in Figure 10.
 817
 818
 819
 820
 821
 822
 823
 824
 825
 826

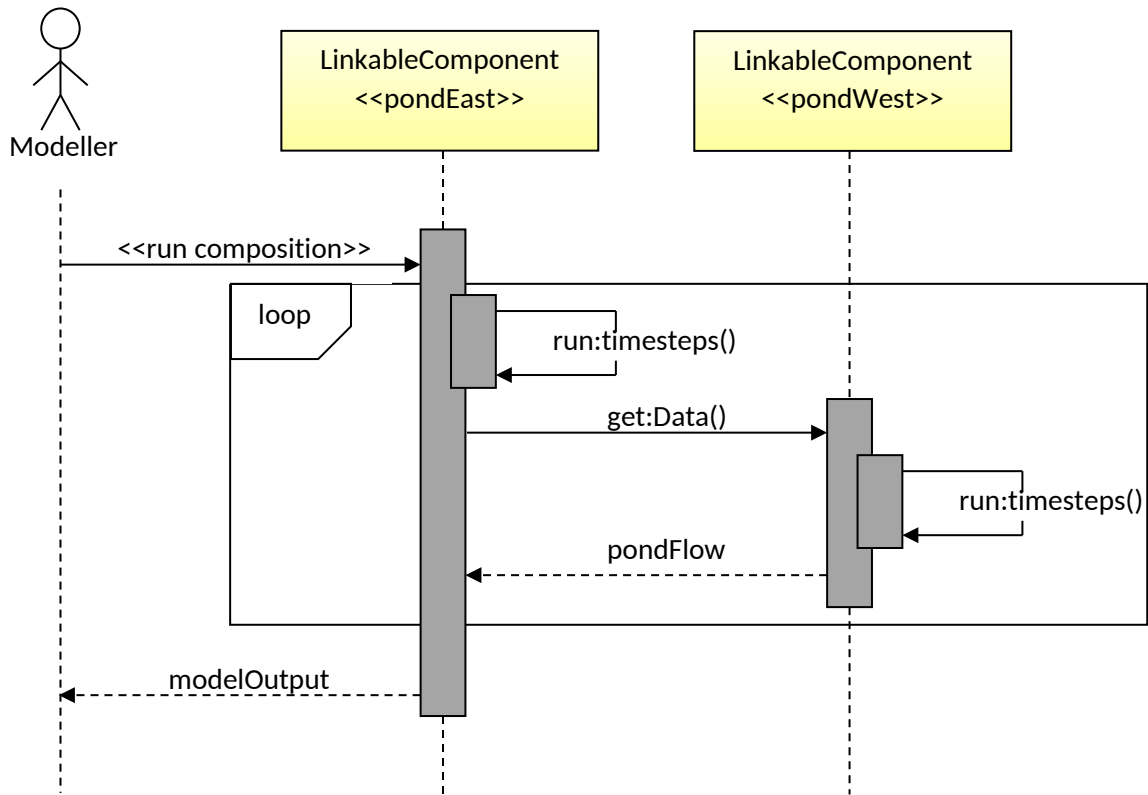


Figure 10: Sequence diagram for run and timestepping elements of simple pond composition

Figure 11 shows this composition in the Pipistrelle GUI where pondEast is attached to the trigger, denoted by the small OpenMI logo. If there are any adaptors present across the link between the two pond components, then this is shown by the bold connecting arrow. Clicking on this arrow will show the attributes of the adaptor(s).

886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944

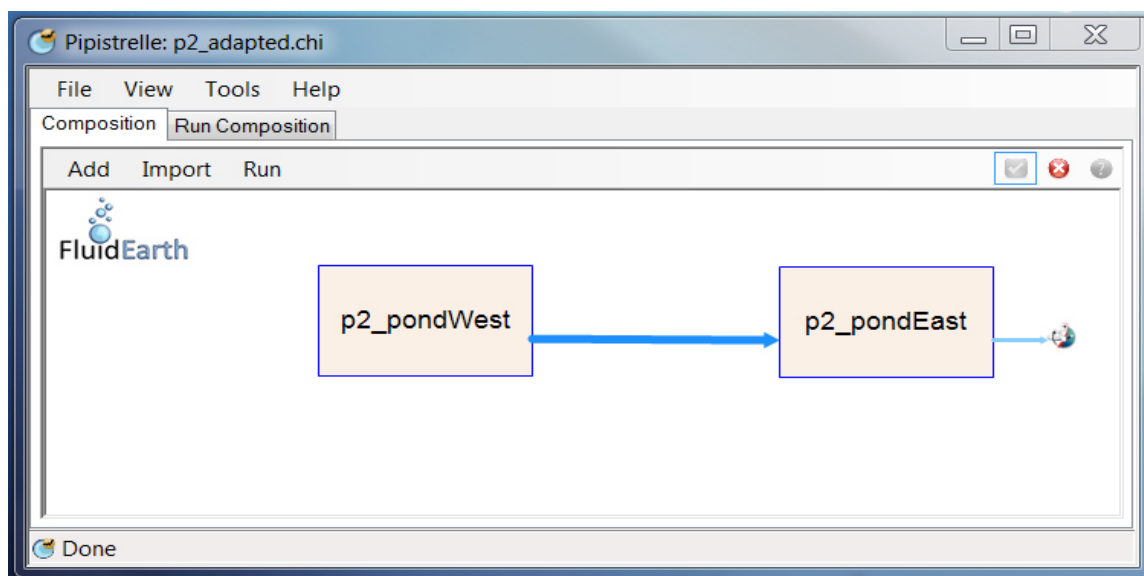


Figure 11: The simple pond composition shown in the Pipistrelle GUI

Pipistrelle will also produce a run log, shown in Figure 12. The run log does not give the model outputs, which are produced in additional files, it merely reports the status of the components and the composition itself.

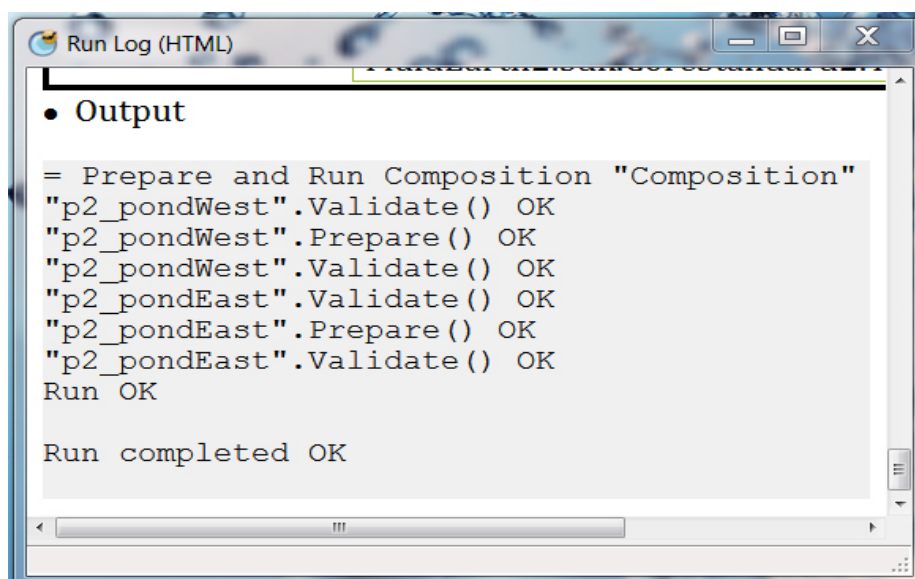


Figure 12: Pipistrelle GUI run log for the simple pond composition

Examples of OpenMI 2.0 Compositions

River channel – floodplain interactions for flash flooding in Genoa

Harpham et al. (2016) includes an OpenMI composition modelling a flash flood which occurred in Genoa in October 2014. An almost identical event had occurred three years earlier in November 2011 when a third of the average annual rainfall fell in six hours. Both events included one-or-more fatalities and a vast amount of damage was caused. An OpenMI 2.0 composition was used by HR Wallingford to model the downstream hydraulics which studied the interaction between the river channel and the flood plain. Along the river side, flood water passed from the river into the town and also from the streets of the town back into the river. Thus, a two-way connection was required

between the numerical model simulating the flow in the channel overtopping to the streets and also the numerical model simulating the movement of water through the streets and then back into the channel.

The composition was put together as part of the Distributed Computing Infrastructure for Hydro-Meteorology (DRIHM) project (D'Agostino et al. 2014, Danovaro et al., 2014, D'Agostino et al., 2015). The FluidEarth implementation of OpenMI 2.0 was used to build the OpenMI components and then assemble and run the composition, which can be seen in Figure 13.

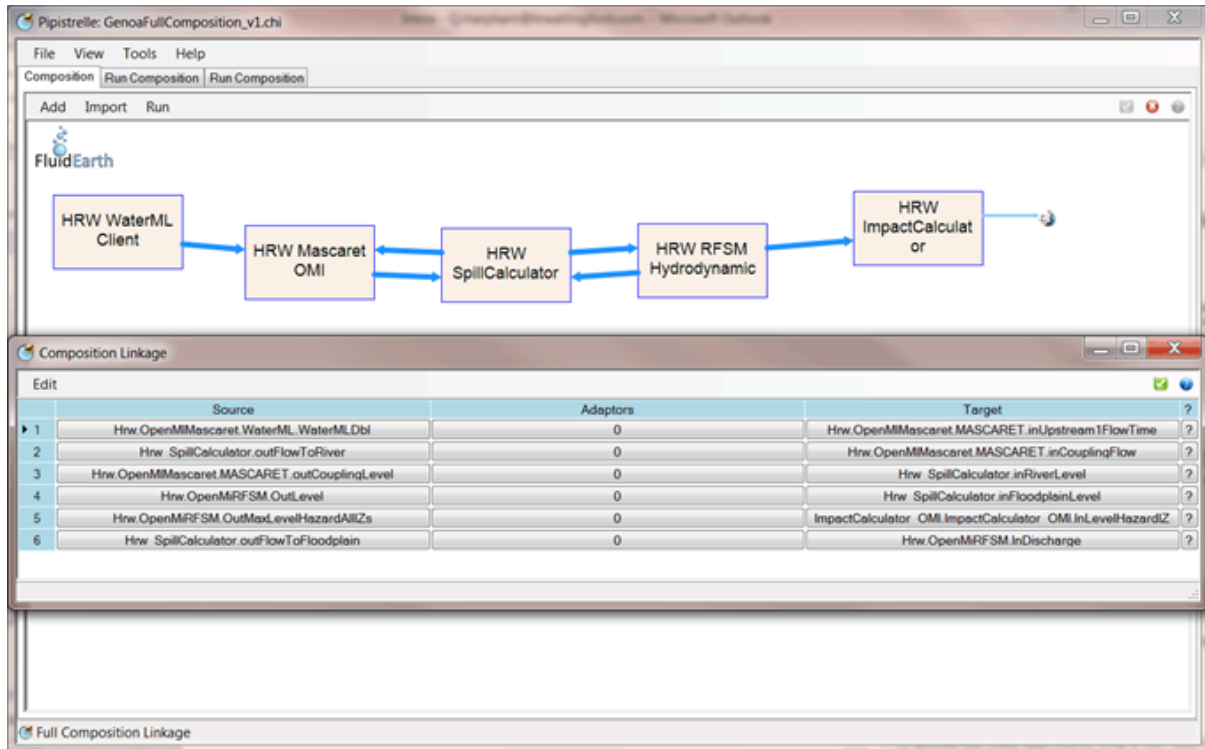


Figure 13: The Pipistrelle GUI showing the OpenMI composition studying the Genoa flash floods of November 2011 and October 2014

The Linkable Components used in the composition are as follows (more detail is given in Harpham et al. 2016):

- **HRW WaterML Client:** A data ingestion translating the upstream flow-time boundary condition from the WaterML2 standard into that required by the following OpenMI component, in a one-way connection.
- **MASCARET** (Goutal et al. 2012): simulating the flow in the river channel.
- **HRW Spill Calculator:** Converting water level to and from flow. This component could also be an adaptor between the MASCARET and RFSM components.
- **HRW RFSM Hydrodynamic:** A flood spreading model calculating flow across the flood plain, in this case, the streets of Genoa (Jamieson et al. 2012).
- **HRW Impact Calculator:** a tool used to estimate the impact of flooding on buildings and agricultural land and predicted Loss of Life using the Risk To People method published in the DEFRA report FD2317 (Ramsbottom et al., 2003).

Use of the OpenMI 2.0 composition enabled the flood event to be accurately simulated with water flowing to and from the flood plain. Existing, well established numerical models were able to be linked together into a composition thereby providing a new capability in simulating such events. As

1004
1005
1006 the water flows through the streets the damage to buildings and hazard to people is calculated by
1007 the Impact Calculator Linkable Component. A single frame of the results animation is given in Figure
1008 14, showing the flow of water through the city as the event progressed. There was a large exchange
1009 of water between the channel and the flood plain in the area highlighted as the channel proceeds
1010 under the railway line. A high hazard was identified at this point, which is, sadly, where one of the
1011 fatalities occurred. The total damage for the 2014 flood was calculated to be €205 million (at 2014
1012 prices). This was within 2.5% of the unofficial locally calculated estimates.
1013
1014



1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034 Figure 14: Frame of animation of water flow through Genoa during the flash flood of October 2014

1035 1036 **Thames integrated model**

1037 BGS's Thames integrated model (TIM) links the detailed geology of the Thames catchment with
1038 groundwater and surface water hydrology including rainfall, runoff, recharge, and to a limited extent
1039 the source and resource management applied in the Thames catchment. There has been particular
1040 emphasis on the geology of the greater London area (Royse et al., 2012). The project brings together
1041 a unique combination of geological, hydrogeological, environmental and socio-economic challenges
1042 that are intrinsically linked and impacted by climate change. To address these challenges requires
1043 fully attributed 3D models that incorporate information and processes from all of these disciplines
1044 so that accurate representations, simulations, forecasts and predictions can be made. These
1045 forecasts and predictions are required to enable informed decision making and planning for
1046 sustainability. In particular, integrated modelling is being applied to understand the interaction
1047 between rainfall evaporation and runoff, together with river flows for the Thames and its tributaries,
1048 with groundwater flow.
1049
1050

1051 The importance of interaction between the surface water system and abstraction has been
1052 demonstrated by using the TIM linked model composition developed by Mansour et al. (2013). A
1053 model composition which allows an appropriate representation of the hydrological system has been
1054 developed and is shown diagrammatically in Figure 15. Two groundwater models have been
1055 developed: one of the chalk using ZOOMQ3D (Jackson and Spink, 2004) and another of the Jurassic
1056 limestones using BGS GW (Mansour et al., 2013). The former is a distributed groundwater flow model
1057 and the latter is a semi-distributed lumped parameter model that can be used to simulate
1058
1059
1060
1061
1062

1063
1064
1065 groundwater behaviour. The boundaries of both models are shown on Figure 16 and it should be
1066 noted that for the chalk model these extend outside of the Thames Basin. This is to ensure that
1067 sensible groundwater flow boundaries are defined so as not to erroneously calculate baseflow to
1068 rivers whose surface catchments are different to groundwater catchments and unduly affect the
1069 impacts of groundwater abstractions.
1070

1071
1072 The two groundwater models are linked via a river model (MCRouter) developed using the
1073 Muskingham-Cunge approach (Chadwick and Morfett, 1986). A simple hydraulic river model was
1074 chosen to ensure the composition can be tested before including a more complex river model.
1075 These three models are driven by run-off and recharge generated by the recharge model ZOODRM
1076 (Mansour and Hughes, 2004; Hughes et al., 2008). The run-off is routed to the river model whilst the
1077 recharge is passed to the two groundwater flow models (Figure 15). As described above, the Jurassic
1078 limestones and chalk aquifer are not linked via the sub-surface, only via interaction with the River
1079 Thames and its tributaries. The two groundwater models and the river model are, therefore,
1080 dynamically linked by the model linkage standard OpenMI (Gregersen et al., 2007). This linkage has
1081 been facilitated by using the Fluid Earth software development kit (SDK) and the associated editor,
1082 Pipistrelle, to construct compositions (Harpham et al., 2014).
1083

1084
1085 The model composition has been applied to a hypothetical situation to test the applicability of the
1086 models and the benefit is accrued from linking them. A hypothetical groundwater abstraction of 150
1087 Mlday⁻¹ has been used to investigate the impacts of groundwater abstraction on the flows in the
1088 River Thames. The groundwater abstraction was simulated from a point on the banks of the River
1089 Thames (labelled BH on Figure 16). This abstraction was linked to river flow as measured at a
1090 downstream gauging station (labelled GS-B on Figure 16). To investigate the effect of different
1091 management regimes two scenarios were simulated: One with a fixed groundwater abstraction of
1092 150 Ml/d and the other with an abstraction that varies from 50 to 150 Ml/d depending on river flow
1093 at GS-B (Figure 16). The relationship between river flow and magnitude of groundwater withdrawal
1094 was achieved by the inclusion of an abstraction management component in the composition which
1095 modified groundwater abstraction during runtime. This component related groundwater abstraction
1096 by a simple rule and was included in the composition. The abstracted water is returned to the river
1097 some 35 km upstream (labelled GS-A on Figure 16) of the groundwater abstraction as this represents
1098 typical water use, from abstraction, by the city of Oxford and after use discharge to the sewerage
1099 system and hence finally sewage effluent returned to the river.
1100

1101
1102 The model composition was simulated for a 30 year period and groundwater heads and the river
1103 hydrograph downstream of the groundwater abstractions plotted (see Figure 3 and 4 in Mansour et
1104 al., 2013). During conditions of low flow, i.e. during the 1975/6 drought the model simulation
1105 resulted in a river flow that was lower for the scenario with fixed abstraction than for the scenario
1106 where groundwater abstraction decreased. However, as groundwater abstraction was reduced,
1107 groundwater levels surrounding the abstraction were higher. Whilst the variable abstraction
1108 reduced the impact on groundwater heads the overall impact on river flows increased due to the
1109 reduction in return flows to the River Thames. Whilst this is a hypothetical example, the results are
1110 contrary to expectation and the utility of the modelling composition in the development of
1111 management policies for droughts and its potential for other scenarios was demonstrated.
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121

1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180

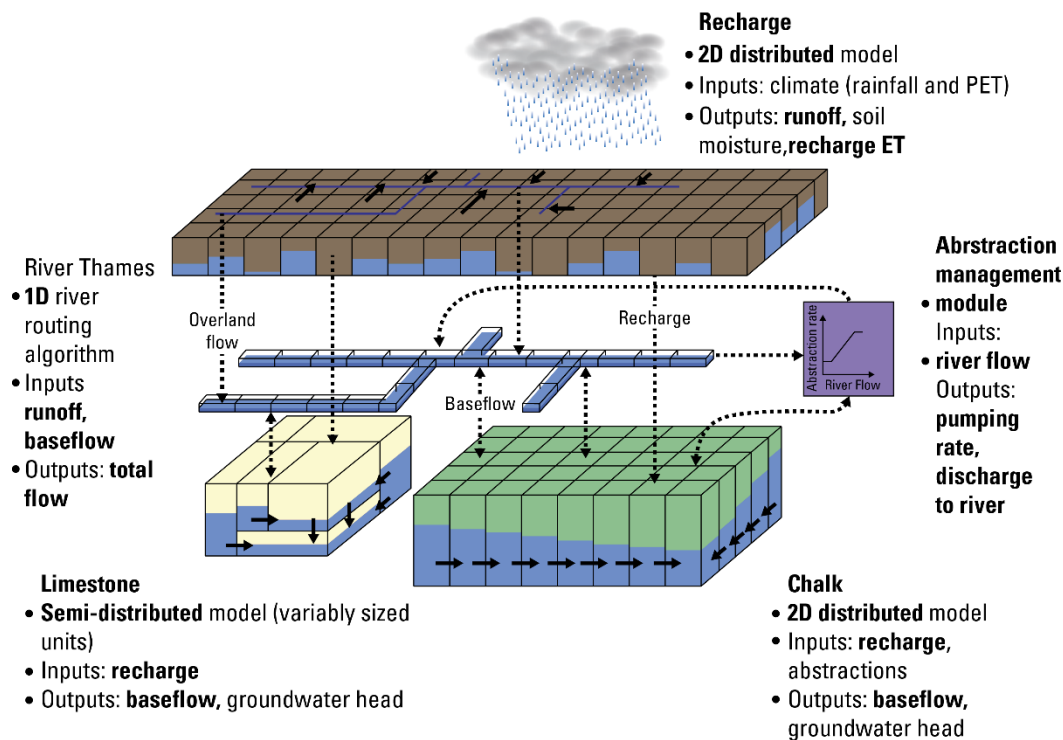


Figure 15: Representation of the hydrological system in the Thames integrated model

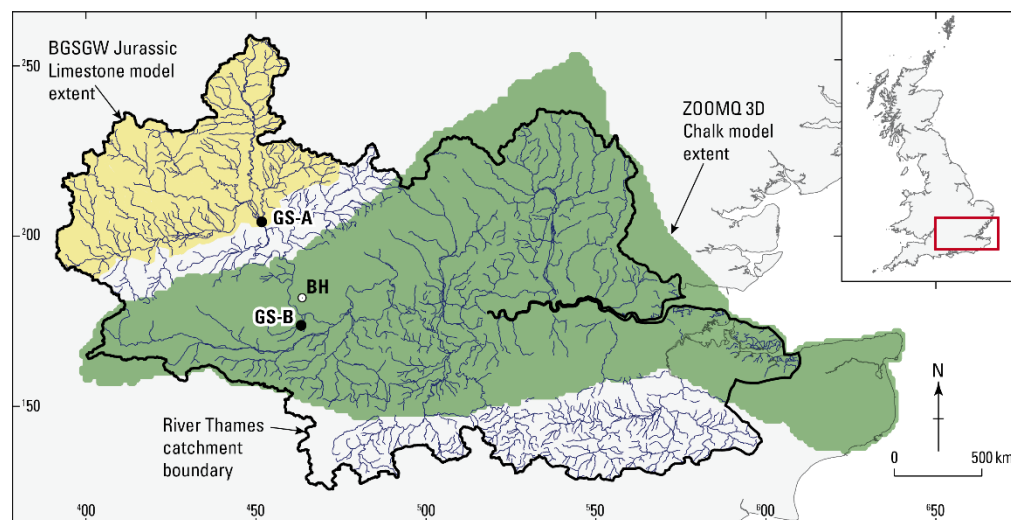


Figure 16: Individual model extents in the Thames integrated model

Conclusions

Sharing the same purpose and motivation as its predecessor, OpenMI 2.0 has a number of similarities and a number of differences to the earlier version of the standard, OpenMI 1.4. Both have been designed to be as open and flexible as possible and allow numerical models to be coupled easily in a manner which allows two-way passage of data between them whilst they are in the process of running. In this way, the numerical models can influence each other. The basic constructs

1181
1182
1183 are the same: compliant numerical models become Linkable Components offering and receiving
1184 Exchange Items to and from each other. No assumptions are made about the spatial structures
1185 involved with both versions of the standard offering low level spatial constructs, rather than
1186 assuming that Grids or Meshes will be used. Similarly, parameters are broken down into their
1187 Dimension Base offering a low level definition, designed to reduce the probability of errors in the
1188 passage of data along two connected Exchange Items.
1189

1190 Unlike version 1.4, OpenMI 2.0 does not assume that the models used will be stepping through time
1191 as they run – although this is most often the case – and offers timestepping as an extension to the
1192 base standard. In order to offer the modeller maximum flexibility, version 2.0 defines any adaptation
1193 of outputs or inputs as a separate entity (an adaptor) to the Linkable Components rather than
1194 embedding such an adaptation in one of the Linkable Components. Also, the passage of data
1195 between Linkable Components is more efficient in version 2.0 of the standard.
1196
1197

1198 There is a trade-off between flexibility and performance. Coupling numerical models in this highly
1199 flexible way can therefore cause performance degradation. The run time of a composition of two
1200 connected numerical models will often be more than the sum of the independent run times. This
1201 degradation varies depending on the implementation. In the worst cases, this is the sum of the two
1202 run times plus the communication time passing data plus the extra run time caused by breaking up
1203 the run of each model; at best the composition runs for slightly longer than the longer of the two
1204 run times. Although individual numerical models can still benefit from parallel high performance
1205 computing (HPC) architectures while running as linkable components, OpenMI 2.0 doesn't include a
1206 specific support for running model compositions on HPC. However, such a specification has been
1207 explored by Buahin and Horsburgh (2018), offering HydroCouple interface definitions for
1208 consideration in future versions of the OpenMI standard.
1209
1210

1211 The integrated modelling solution offered by OpenMI 2.0 has been demonstrated to work
1212 successfully and offer numerical modellers a valid method for the two-way coupling of numerical
1213 models. In many ways, any alternative solution would have to function similarly: numerical models
1214 would need to offer clear outputs and clear inputs; control would need to be passed to an external
1215 entity so that a controllable composition is possible; some sort of adaptation would be required
1216 across connections to deal with any inherent differences between candidate inputs and outputs.
1217 Thus OpenMI 2.0 offers a natural and intuitive method for integrating numerical models.
1218

1219 OpenMI has originated from the domain of hydraulic modelling. It has been devised to integrate
1220 such things as models of drainage systems with models of river systems; models of groundwater
1221 with models of surface water. However, it can be observed that similar concepts exist in other
1222 spheres. A water supply network is a pumped system of pipes under pressure, entirely analogous to
1223 a biological bloodstream. An internal combustion engine cylinder also contains a set of separate yet
1224 integrated processes as the gases are introduced, ignited and exhausted from a cylinder. As such,
1225 OpenMI 2.0 can be used in all circumstances where it is necessary to exchange data between
1226 numerical models, in particular when the models must be allowed to influence each other as they
1227 proceed.
1228
1229

1230 Acknowledgements

1231 OpenMI 2.0 and the worked examples given here are the result of over a decade of research by a
1232 wide variety of individuals. The projects include:
1233

- 1234 • HarmonIT (EC contract: EVK1-CT-2001-00090) and OpenMI-Life (Grant agreement number
1235 LIFE06 ENV/UK/000409));
1236
1237
1238
1239

- 1240
1241
1242
1243
1244
- DRIHM (EC Grant Agreement 283568) and DRIHM2US (EC Grant Agreement 313122);
 - FluidEarth, HR Wallingford.

1245 The authors would also like to acknowledge the technical contribution from the OpenMI Association
1246 Technical Committee (OATC) and support throughout the standardisation process from the Open
1247 Geospatial Consortium (OGC).
1248

1249 More recent incorporated research into spatio temporal structures has been conducted as part of
1250 the UKSA IPP 'D-MOSS' project building and implementing a dengue fever forecasting system for
1251 Vietnam ([https://www.gov.uk/government/news/new-projects-see-uk-space-firms-tackle-](https://www.gov.uk/government/news/new-projects-see-uk-space-firms-tackle-southeast-asian-challenges)
1252 [southeast-asian-challenges](https://www.gov.uk/government/news/new-projects-see-uk-space-firms-tackle-southeast-asian-challenges)).
1253
1254

1255 References

1256 Anastas, P. 2010 Agency Priority. EPA Office of Research and Development, Washington, D.C.
1257

1258 Buahin, C. and Horsburgh, J. 2018. Advancing the Open Modelling Interface (OpenMI) for Integrated
1259 Water Resources Modeling, *Environmental Modelling & Software*, 108, pp 133-153. doi
1260 10.1016/j.envsoft.2018.07.015.
1261

1262 Chadwick, A. and Morfett, J., 1986. *Hydraulics in Civil Engineering*. Allen and Unwin (Publishers) Ltd,
1263 London, UK.
1264

1265 D'Agostino, D., Clematis, A., Galizia, A., Quarati, A., Danovaro, E., Roverelli, L., Zereik, G.,
1266 Kranzlmuller, D., Schiffers, M., gentschen Felde, N., Straube, C., Caumont, O., Richard, E., Garrote, L.,
1267 Harpham, Q., Jagers, B., Dimitrijevic, V., Dekic, L., Parodi, A., Fiori, E. and Delogu, F., 2014 The DRIHM
1268 project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological
1269 research, SC '14: Proceedings of the International Conference for High Performance Computing,
1270 Networking, Storage and Analysis. pp 536-546, IEEE Press Piscataway, NJ, USA © 2014. doi
1271 10.1109/SC.2014.49.
1272

1273 D'Agostino, D., Danovaro, E., Clematis, A., Roverelli, L., Zereik, G., Parodi, A. and Galizia, A. 2015.
1274 Lessons learned implementing a science gateway for hydrometeorological research. *Concurrency
1275 and Computation: Practice and Experience*.
1276

1277 Danovaro, E., Roverelli, L., Zereik, G., Galizia, A., D'Agostino, D., Quarati, A., Clematis, A., Delogu, F.,
1278 Fiori, E., Parodi, A., Straube, C., Felde, N., Harpham, Q., Jagers, B., Garrote, L., Dekic, L., Ivkovic, M.,
1279 Richard, E. and Caumont, O. Setup an hydro-meteo experiment in minutes: the DRIHM e-
1280 infrastructure for hydrometeorology research. *Proceedings e-Science 2014: 10th IEEE International
1281 Conference on e-Science*, Guarujá, SP, Brazil, October 20-24, 2014.
1282

1283 Gregersen, J. B., Gijssbers, P. J. A. and Westen, S. J. A. 2007. OpenMI: Open Modelling Interface,
1284 *Journal of Hydroinformatics* Issue 9(3) 2007 pp 175-191.
1285

1286 Goutal, N., Lacombe, J-M., Zaoui, F. and El-Kadi-Abderrezzak, K. 2012. MASCARET: a 1-D Open-
1287 Source Software for Flow Hydrodynamic and Water Quality in Open Channel Networks, *River Flow
1288 2012 - Murillo (Ed.)*, pp. 1169-1174.
1289

1290 Harper, A., Barnes, J. Cleverley, P., Harpham, Q. and Hummel, S. 2016. FluidEarth,
1291 <https://sourceforge.net/projects/fluidearth/>, accessed 3rd April 2019.
1292
1293
1294
1295
1296
1297
1298

- 1299
1300
1301 Harpham, Q. K., Cleverley, P. & Kelly, D. 2104 The Fluid Earth 2 implementation of OpenMI 2.0. J.
1302 Hydroinform. 16(4), 890–906.
1303
- 1304 Harpham, Q., Lhomme, J., Parodi, A., Fiori, E., Jagers, B. and Galizia, A. 2016. Using OpenMI and a
1305 Model MAP to Integrate WaterML2 and NetCDF Data Sources into Flood Modeling of Genoa, Italy.
1306 JAWRA Journal of the American Water Resources Association (2016).
1307
- 1308 Harpham, Q.K. 2018. Using spatio-temporal feature type structures for coupling environmental
1309 numerical models to each other and to data sources, under review for publication, Open University,
1310 Milton Keynes, UK.
1311
- 1312 Harpham, Q. and Danovaro, E., 2015. Towards standard metadata to support models and interfaces
1313 in a hydro-meteorological model chain. Journal of Hydroinformatics, 17(2), pp.260-274.
1314
- 1315 Hughes, A. G., Mansour, M. M., & Robins, N. S. 2008. Evaluation of distributed recharge in an upland
1316 semi-arid karst system: the West Bank Mountain Aquifer, Middle East. Hydrogeology Journal, 16(5),
1317 845-854.
1318
- 1319 Iwanaga, T., Rahman, J., Partington, D., Croke, B. and Jakeman, A. 2018. 'Software Development
1320 Practices in Integrated Environmental Model Development', Proc. 9th International Congress on
1321 Environmental Modelling and Software, Fort Collins, Colorado, USA. Mazdak Arabi, Olaf David, Jack
1322 Carlson, Daniel P. Ames (Eds.).
1323
- 1324 Jackson C.R.; Spink A.E.F., 2004. User's manual for the groundwater flow model ZOOMQ3D.
1325 Nottingham, UK, British Geological Survey, 107pp. (IR/04/140) - <http://nora.nerc.ac.uk/11829/>
1326
- 1327 Jamieson, S., Lhomme, J., Wright, G. and Gouldby, B. 2012. Highly efficient 2D inundation modelling
1328 with enhanced diffusion-wave and sub-element topography. Proc. Inst. Civ. Eng. Wat. Man. 165 (10):
1329 581–595.
1330
- 1331 Laniak, G.F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N.,
1332 Blind, M. and Peckham, S., 2013. Integrated environmental modeling: a vision and roadmap for the
1333 future. Environmental Modelling & Software, 39, pp.3-23.
1334
- 1335 Mansour, M.M.; Hughes, A.G. 2004 User's manual for the distributed recharge model ZOODRM.
1336 British Geological Survey, 61pp. (IR/04/150) - nora.nerc.ac.uk/12633/
1337
- 1338 Mansour, M., Mackay, J., Abesser, C., Williams, A., Wang, L., Bricker, S., & Jackson, C. 2013.
1339 Integrated Environmental Modeling applied at the basin scale: linking different types of models
1340 using the OpenMI standard to improve simulation of groundwater processes in the Thames Basin,
1341 UK. In: MODFLOW and More 2013: Translating Science into Practice, Colorado, USA, 2-5 June 2013.
1342 (nora.nerc.ac.uk/501789/)
1343
- 1344 Meiburg, S. in EPA 2008 Integrated Modeling for Integrated Environmental Decision Making. EPA-
1345 100-R-08-010. US Environmental Protection Agency, Office of the Science Advisor, Washington, DC.
1346
- 1347 Moore, R. V., & Hughes, A. G., 2017. Integrated environmental modelling: achieving the vision.
1348 Geological Society, London, Special Publications, 408(1), 17-34.
1349
- 1350 Nativi, S., Mazzetti, P., & Geller, G. N., 2013. Environmental model access and interoperability: The
1351 GEO Model Web initiative. Environmental Modelling & Software, 39, 214-228.
1352
1353
1354
1355
1356
1357

1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416

OGC OpenMI. 2014. OGC Open Modelling Interface - Interface Standard OGC 11--014r3 version 2.0. Open Geospatial Consortium, <http://www.opengis.net/doc/IS/openmi/2.0GC>, accessed 26th February 2019.

Ramsbottom D., Floyd P. and Penning-Rowell E. 2003. Flood Risks to People Phase 1 R&D Technical Report FD2317, ISBN 1844321355.

Royse, K. R., de Freitas, M., Burgess, W. G., Cosgrove, J., Ghail, R. C., Gibbard, P., ... & Skipper, J. (2012). Geology of London, UK. Proceedings of the Geologists' Association, 123(1), 22-45.

Sophocleous, M. A., Koelliker, J. K., Govindaraju, R. S., Birdie, T., Ramireddygari, S. R., & Perkins, S. P., 1999. Integrated numerical modeling for basin-wide water management: The case of the Rattlesnake Creek basin in south-central Kansas. Journal of Hydrology, 214(1), 179-196.

Voinov, A. 2010 Model integration and the role of data. Environmental Modelling & Software 25, 965-969.