Applying the Open Modelling Interface (OpenMI)

Dudley, J[1], W Daniels[2], PJA Gijsbers[3], D Fortune[2], S Westen[2], and JB Gregersen[4]

[1] WRc plc, UK

[2] Wallingford Software Ltd, UK

[3] WL/Delft Hydraulics, The Netherlands, peter.gijsbers@wldelft.nl

[4] DHI Water and Environment, Denmark

Keywords: *OpenMI, model integration, data exchange, interface specification*

EXTENDED ABSTRACT

Catchment-level modelling has been regarded, since at least the middle 1980s, as a necessary part of providing integrated solutions for improving water quality in receiving waters. However, the tools that have been available to support this level of modelling have not made such modelling convenient. Typically there has been no support for data exchange between different vendors, and different classes of model (e.g., rivers, sewers and sewage works) have used different water quality parameters in their assessment. The purpose of the OpenMI has been to provide a first level of greater ease of data exchange between different programs, by removing the barrier of exchanging data between different programs. The OpenMI has also provided mechanisms to simplify exchanging quantities between different programs – for example, should one program expect flow in US units, and another in cumecs, simple transformations can be defined to ensure that each program can convert the data from the other to the required units. A relatively large-scale application of the OpenMI in a model catchment is described here, to better illustrate some of the benefits and modelling issues that are introduced by use of the OpenMI.

This application links rainfall, runoff, river, sewer, sewage works and lake models from three different companies, using two different sets of flow units and three different water quality parameter sets (BOD or COD, and how COD is partitioned). The OpenMI provided sufficient flexibility to permit this connection, handling much of the data transformations within the OpenMI environment without needing explicit user intervention.

It is concluded that the OpenMI has facilitated linking many disparate water-cycle programs. The focus of OpenMI (i.e. data exchange at the engine /component level) and the current state of adoption by the software community require the usage of a straightforward editor to define the linkage outside the user interface environments of the various systems. This focus has kept the conceptual overhead low, as each program in the catchment model is manipulated individually, so that the domain experts do not need to learn a new interface. In a similar way, the results of the integrated computation need to be inspected through the individual user interfaces of the various models incorporated.

The OpenMI provides a feasible for solution for the IT-communication problem. However, it does not solve all problems. As each program exchanges data with a program from a different model developer there has been the need to communicate to ensure that the semantics are clear, that the connection points are correct, and that the data being transferred is mapped correctly – e.g., agreeing on transformation protocols between COD from the sewage models to BOD in the lake model. This communication has been a source of difficulty with previous efforts at integrated catchment modelling, and while OpenMI can simplify the technical aspects, it does not address these human issues.

Finally, the analysis of simulation results requires further communication between the domain experts. The OpenMI thus allows problems to be tackled in a more integrated manner, but does not remove the constraints of ensuring communication between the different technical work areas in understanding the total output. As well as needing the traditional local areas of expertise – for example, river, sewerage and sewage treatment works modellers – there has also been the greater need for a new area of expertise, providing the catchment-level understanding of the whole problem. The OpenMI, by providing a framework for these different areas to better integrate the modelling work, has concomitantly provided a social framework to encourage the individual teams to work together to provide better modelling data for the decision makers.

1. INTRODUCTION

The European Union (EU) has introduced the Water Framework Directive (WFD) as a legislative driver to encourage catchment-wide decisions on water quality, covering not only organisational (e.g. water and sewerage providers) and local boundaries (e.g. different towns along a river) but also trans-national boundaries.

One expectation of the WFD is that there will be greater use of modelling tools to help provide the integration of data sources and environmental effects across these large catchments. Historically the models have developed as stand-alone tools, making data exchange, let alone integration, at best difficult. The EU therefore funded a project to develop a new interface mechanism, the Open Modelling Interface (OpenMI) to provide a means for software to be developed in such a manner as to support data exchange and communication. OpenMI-enabled software tools will allow existing models of parts of a catchment to be integrated into a larger, more complete, model of the whole catchment. The technical details of OpenMI are presented in Gijsbers & Gregersen,(2005).The OpenMI also provides mechanisms to address additional problems with such integration, such as spatial equivalency (when one model uses, say, a one-dimensional representation while another uses, perhaps, a two-dimensional version), datum levels (while the water levels at model boundaries should match, the apparent water levels will be affected by the choice of the base level) and water quality parameters (some models use BOD, others COD, and the fractionation of COD, for example, may not be defined in the same way between models which apparently both use COD as their basis.)

This paper illustrates two points:

- How OpenMI can make integrated catchment modelling easier, as regards the mechanics of connecting different programs together and exchanging data between these programs; and

- Bring out that the technology is not sufficient by itself; the different skills available through different modelling areas (e.g., sewers, rivers) must cooperate in ensuring that the different programs connect in a manner that will solve the problem.

There is a further level of involvement that has not been addressed, but which is also important, namely that of the decision makers in making use of the output from these programs in reaching environmental solutions.

2. A SIMPLE CATCHMENT MODEL

This paper introduces a simple catchment model constructed through integrating sub-models from different software providers. While the catchment is artificial, constructed to demonstrate the applicability of the OpenMI, it is loosely based on our collective experience of catchments, and of issues concerning model integration that we wish to address within this paper.

3. MODELS AND CONNECTIONS

The integrated model is represented in Figure 1, showing a river system with runoff, connected sewer systems and sewage works, feeding a lake, with the lake discharging at one end. Various water-cycle models are used to represent the

Table 1 Programs used in the catchment model

Link	From	To	Uni or bidirectional?	Water quality?
A	SOBEK-RR (Rainfall / Runoff)	SOBEK-CF (Channel Flow)	⇨	
B	SOBEK-SF (Sewer Flow)	SOBEK-CF (Channel Flow)	⇨	
C	HYMOS Database (Rainfall)	SOBEK-RR (Rainfall / Runoff)	⇨	
D	SOBEK-CF (Channel Flow)	Infoworks RS (Channel Flow)	⇔	
E	Infoworks CS (Sewer Flow)	STOAT (Wastewater treatment works)	⇨	✓
F	STOAT (Wastewater treatment works)	Infoworks RS (Channel Flow)	⇨	✓
G	Infoworks CS (Sewer Flow)	Infoworks RS (Channel Flow)	⇨	✓
H	Infoworks RS (Channel Flow)	SULIS (Lake)	⇔	✓
I	Infoworks CS (Sewer Flow)	STOAT (Wastewater treatment works)	⇨	✓
J	STOAT (Wastewater treatment works)	SULIS (Lake)	⇨	✓
K	SOBEK-RR (Rainfall / Runoff)	SULIS (Lake)	⇨	
L	SULIS (Lake)	Infoworks RS (Channel Flow)	⇔	✓

complete catchment, and the details of the models and their connection points are given in Table 1.

This model has been constructed to demonstrate that a complex model, constructed of many programs, can be constructed. Components that would be needed for many such river basin models have been included. Figure 2 provides a depiction of the linkage between these programs.

Many of the programs have not been coupled in the past, and do not have existing interfaces that would allow them to work together. The previous solution has been to edit each program's output to make it compatible with the downstream models. Such sequential oriented method of file linkage would not accommodate any feedback processes that are required to simulate backwater effects properly without the need to move to small timesteps. In addition to the issues of file formats, there is, for water quality, the additional problem of ensuring that consistent information is passed from one program to the next. The lake model, SULIS, uses BOD; the sewage works, sewer and river models are set to use COD. The sewer and sewage works do not use the same fractionating approach for particulate COD, as the sewer model is concerned with the distinction between COD that can settle in the sewer, and COD that, while particulate, will be transported as if soluble. The sewage works models, on the other hand, are concerned with the difference between COD that is immediately taken up by the bacteria, regarded as soluble, and COD that needs to be broken down into soluble COD before it can be utilised by the bacteria, regarded as particulate.

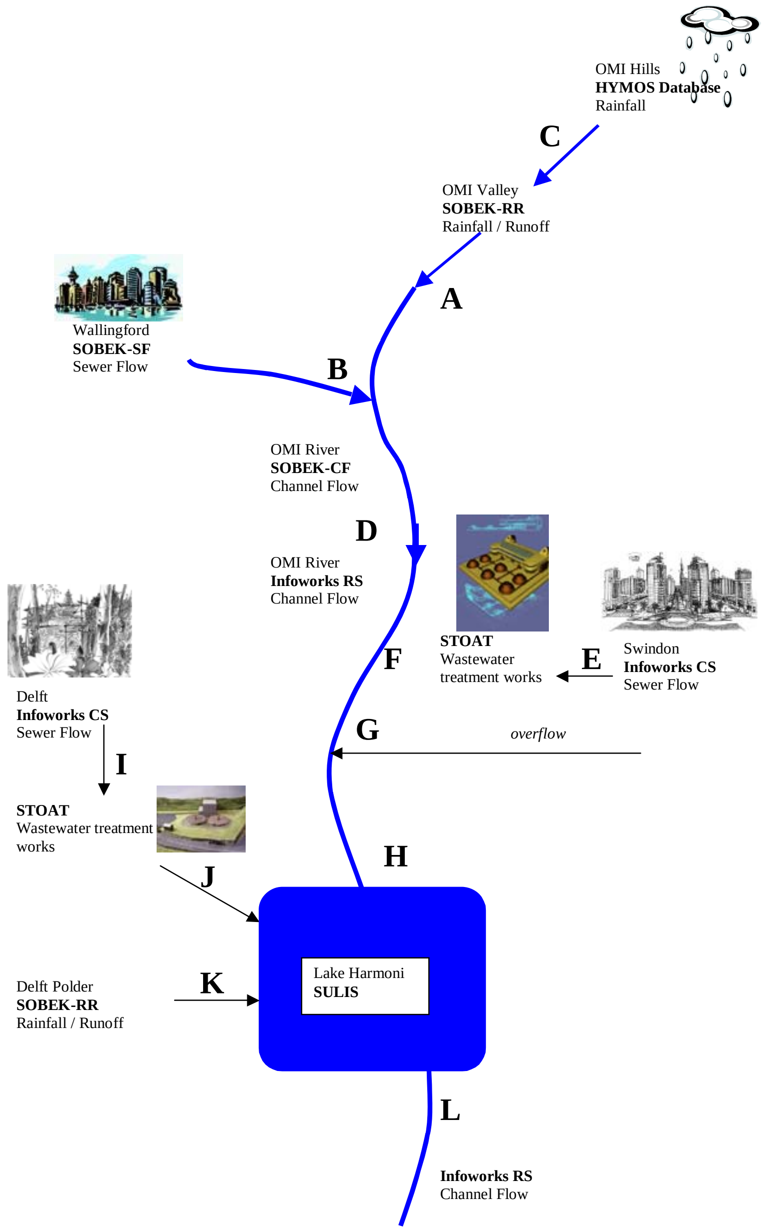

Figure 1 Schematic representation of the integrated catchment model

Previous attempts at creating large integrated models, such as the previous DHI/WRc collaborative project TVP (Taylor *et al.*, 1999), have created GIS-based interfaces that have attempted to simplify visual representation, but not the mechanics of connecting the programs and running them. OpenMI focuses on the linkage issue rather than such software areas as user interfaces. On the contrary, OpenMI does not mandate a user interface and allows users and their suppliers to develop user interfaces for OpenMI or to embed OpenMI functionality in their own products. However, to enable working with OpenMI there is a simple linkage editor provided as part of the C# .Net implementation. This implementation also provides utilities for model wrapping. It is available as open source on sourceforge.net/projects/openmi or through the OpenMI website (www.openmi.org). An example of the linkage editor is shown in Figure 2. This shows the rainfall runoff (RR), sewer (boxes starting CS), river (RS for Wallingford Software's version, CF for Delft's version), sewage works ('Works') and lake ('minibox') models being connected, where there is a sewered connection direct to the lake, a sewer spill point to a river feeding the lake, and the lake discharging to a further river reach. After the programs have been connected the quantities being exchanged are then defined, as illustrated in Figure 3. This displays the available output quantities from a sewer model (CS) and input quantities to a river model (RS). The text in the link box indicates that two connections have already been made, for suspended solids and ammoniacal nitrogen. The legends use whatever names are exported by the programs, which may be cryptic (as seen by suspended solids) or more meaningful (as seen by ammonia).

In the previous TVP programme a GIS-based user interface was available, but this made it more difficult to distinguish between the spatial layout of the models and the boundaries of the programs. The simple approach currently used in the OpenMI demonstrator has been found to enhance understanding of the inter-relationships between the different programs. A GIS-enabled interface that allows switching between the spatial and software views may help in ensuring that

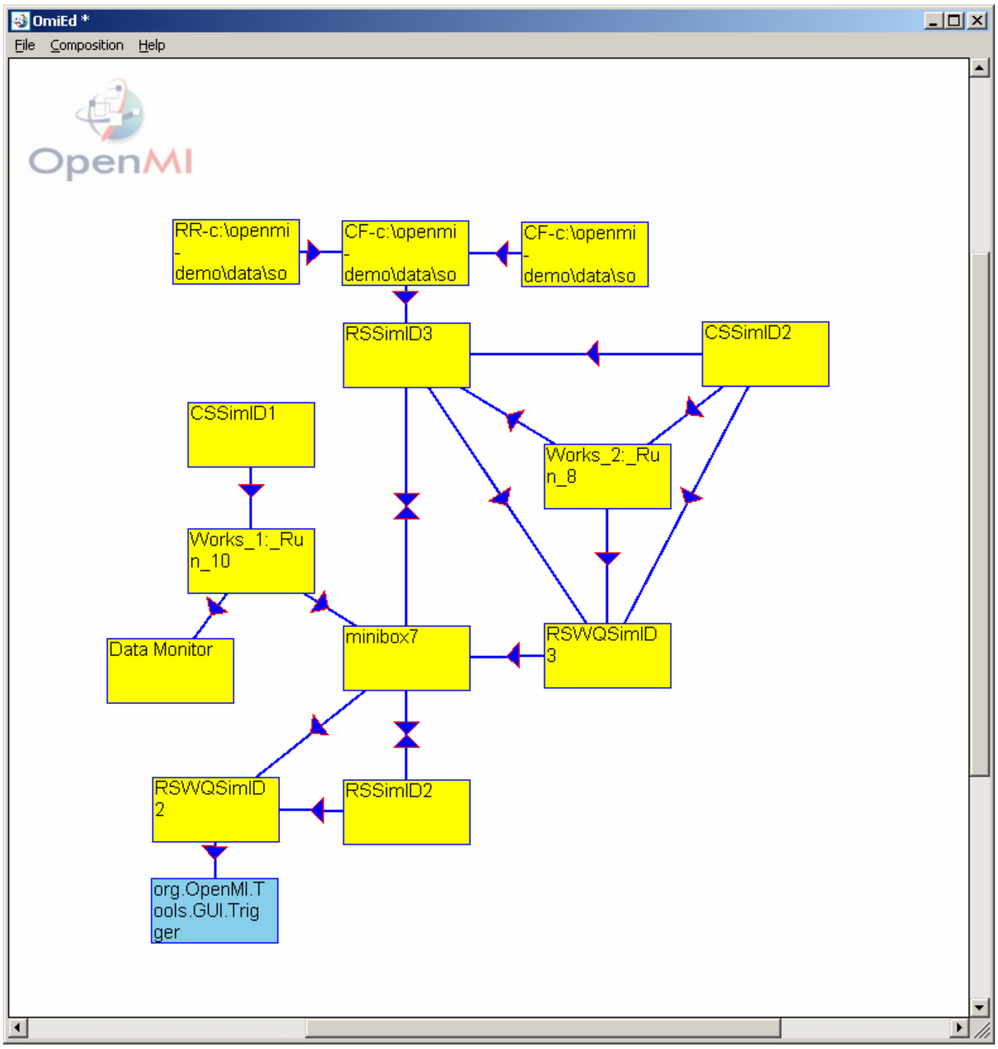

Figure 2 Linking the programs using the OpenMI configuration editor

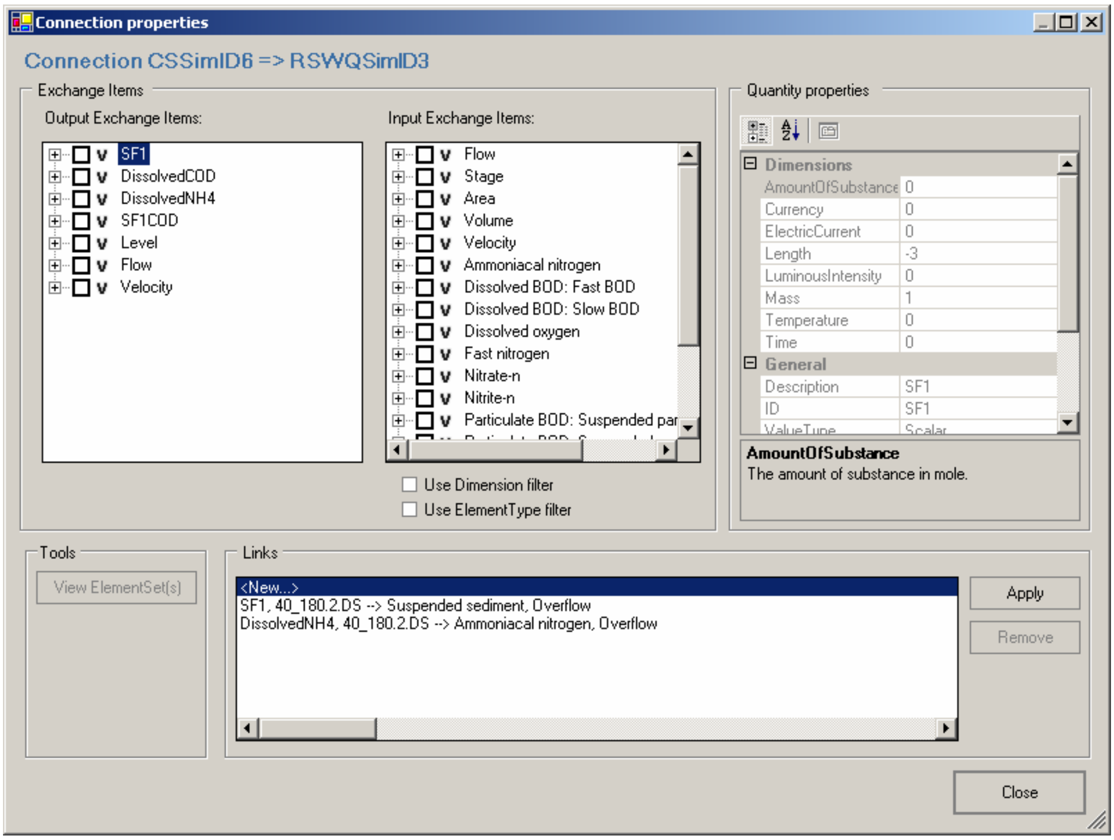

Figure 3 Setting up data exchange using the OpenMI configuration editor. This example shows sewer (CS) to river (RS)

connections are made at the right points, rather than relying upon descriptions (e.g., ‘Swindon overflow’ from the sewer model should connect to ‘CSO spill point’ on the river model.) However, OpenMI also supports mapping of spatial coordinates, so that future iterations of the program may automatically identify the connections through their geographical coordinates, and detect if the connections are not correctly spatially aligned.

4. CONNECTING PROGRAMS

Table 2 provides an overview of the way OpenMI allows model connections to be defined to address the problems described above. The default wrapper implementation for OpenMI accommodates the specification of data operations as presented. The bottom pane of Figure 3 shows the feedback available in the current interface to guide users in ensuring that they are transferring data at the correct location, and mapping quantities correctly. While OpenMI supports transformations, so that quantities can be manipulated at the OpenMI level, this is not as readily available where the mapping is not a simple one-to-one relationship (as, for example, different units for flow), but rather a many-to-one relationship (for example, suspended solids in the target program being the sum of several solids fractions in the source.) This requires that the programs provide support for such mappings, and can be done either at the OpenMI level, or, at a further level within the target program.

The programs have been successfully run coupled together. The OpenMI did not slow the compute speed of the whole catchment model down to any extent, but there was a slow-down caused by the requirement to run the simulation at a small timestep, to capture the dynamics of data exchange across the whole assembly. There was an additional overhead caused by the data exchange between the different programs, but this overhead was present at the Windows level, and is intrinsic to the program communication procedures adopted by the Windows operating system, rather than to those imposed by the OpenMI itself.

Because the OpenMI facilitates linking and running the programs there is no unified reporting mechanism. The output of the various programs stays with those programs, making overlaying results at the connection points (to ensure that data is transferred correctly) a manual exercise, copying the data into, for example, a spreadsheet to allow comparisons. Such an exercise will need to be done only during the early years of using the OpenMI, while confidence is being built up that data is being transferred correctly, and when new programs are migrated to the OpenMI framework.

Without a unified reporting mechanism the OpenMI has made it easier to run these large models, but has not addressed the more important issue of collating all the output into a format that will simplify understanding the whole problem, and reaching appropriate environmental, economic, or engineering decisions.

Table 2 Overview of modelled variables and data operations to match output and input

Link	Output Quantity	Input Quanity	Data operation
A	Runoff (m3/s)	Lateral inflow (m3/s)	-
B	Runoff (m3/s)	Lateral inflow (m3/s)	-
C	Rainfall (mm/s)	Rainfall (mm/h)	Unit conversion
D	Discharge (m3/s) Water level (m above OL)	Flow (m3/s) Stage (m above HL)	- datum offset: -0.03 m
E	Flow COD Suspended solids ….Ammoniacal nitrogen	Flow COD Suspended solids Ammoniacal nitrogen	- -
F	Flow COD … Suspended solids Ammoniacal nitrogen Temperature	Flow COD … Suspended solids Ammoniacal nitrogen Temperature	-
G	Stage (m above HL) Flow COD Suspended solids Ammoniacal nitrogen Temperature	Stage (m above HL) Flow COD Suspended solids Ammoniacal nitrogen Temperature	- - - - - -
H	Stage (m above HL) Flow COD Suspended solids Ammoniacal nitrogen Temperature	Stage (m above HL) Flow BOD Suspended solids Ammoniacal nitrogen Temperature	- - linear conv. 0.75 - - -
I	Flow COD …Suspended solids Ammoniacal nitrogen	Flow COD Suspended solids Ammoniacal nitrogen…	-
J	Flow COD Suspended solids Ammoniacal nitrogen Temperature	Flow BOD Suspended solids Ammoniacal nitrogen Temperature	- linear conv. 0.75 - - -
K	Runoff (m3/s)	Inflow (m3/s)	-
L	Stage (m above HL) Flow BOD Suspended solids Ammoniacal nitrogen Temperature	Stage (m above HL) Flow COD Suspended solids Ammoniacal nitrogen Temperature	-

5. DISCUSSION AND CONCLUSIONS

The OpenMI has facilitated linking many disparate water-cycle programs. Avoiding regarding the purpose of the OpenMI as providing an integrated software suite has kept the conceptual overhead low, as each program in the catchment model is manipulated individually, so that the domain experts do not need to learn a new interface. (Of course, the chosen implementation may include such a new over-arching interface linking the different models – but this is not a requirement of OpenMI, and is instead a policy decision to be made by users or vendors.)

The integration of the different programs into the larger catchment model has required each contributor to describe where the data file containing the submodel connection data is stored on the computer system, and what input and output connections and quantities are involved. As each program is required to exchange data with a program from a different domain expert there has been the need to communicate to ensure that the connection points are correct, and that the data being transferred is mapped correctly – e.g., agreeing on transformation protocols between COD from the sewage models to BOD in the lake model. This communication has been a source of difficulty with previous efforts at integrated catchment modelling, and while OpenMI can simplify the technical aspects it does not address these human issues.

After the catchment model has been run the results from each program need to be analysed in the originating program. The currently available OpenMI implementation provides mechanism for examining exchanged data, but this is not as convenient as the facilities provided in each originating program. Consequently, again, analysing the results in deciding the overall catchment impact has required further communication between the domain experts. The OpenMI allows larger problems to be tackled, but does not remove the constraints of ensuring communication between the different technical work areas in understanding the total output. As well as needing the traditional local areas of expertise – for example, river, sewerage and sewage treatment works modellers – there has also been the greater need for a new area of expertise, providing the catchment-level understanding of the whole problem. This has normally been addressed at a more political level than has been the case for the technical problems, with each technical team attempting to provide its solution to the ultimate decision makers with no interaction, or understanding, of the effects that their solution has on other parts of the complete problem. The OpenMI, by providing a framework for these different areas to better integrate the modelling work, has concomitantly provided a social framework to encourage the individual teams to work together to provide better modelling data for the decision makers.

6. REFERENCES

Taylor, S, B Tomicic, W Williams and K Murray, 1999, "'TVP' – 'ICS' and 'RTC' in Oldham", WaPUG Autumn Meeting, http://www.wapug.org.uk/past_papers/autumn_1999/paper08.pdf

Gijsbers, P.J.A. and Gregersen, J.B., 2005 OpenMI: a Glue for For Integrated Modelling, this conference

7. ACKNOWLEDGMENTS

The authors would like to thank their respective organisations for permission to publish this paper. We would also like to thank the EU for funding the project that led to the development of the OpenMI, and the members of the HarmonIT team for providing tools and support used for this project. The views expressed in this paper are those of the authors, and are not necessarily the views of WRc, Wallingford Software, WL/Delft Hydraulics, DHI Water & Environment , the EU Commission or the HarmonIT partners and consortium.