OpenMI

The OpenMI Document Series

Part A - Scope

For the OpenMI (Version 1.0)

IT Frameworks (HarmonIT)

Contract EVK1-CT-2001-00090

Title	OpenMI Document Series: Part A - Scope for the OpenMI (version 1.0)
Editor	Isabella Tindall, Centre for Ecology and Hydrology, Wallingford, UK
Authors	Roger Moore, Centre for Ecology and Hydrology, Wallingford, UK Peter Gijsbers, WL Delft Hydraulics, Delft, The Netherlands David Fortune, HR Wallingford Group, Wallingford, UK Jan Gregersen, DHI Water and Environment, Hørsholm, Denmark Michiel Blind, RIZA, Lelystad, The Netherlands
Document production	Stephen Morris, Butford Technical Publishing Ltd., Pershore, UK
Current version	V1.0
Date	02/08/2005
Status	Final © HarmonIT
Copyright	All methodologies, ideas and proposals in this document are the copyright of the HarmonIT project participants. These methodologies, ideas and proposals may not be used to change or improve the specification of any project to which this document relates, to modify an existing project or to initiate a new project, without first obtaining written approval from those of the HarmonIT participants who own the particular methodologies, ideas and proposals involved.

Preface

This document provides an overview of the Open Modelling Interface and Environment, the OpenMI, which is being developed by the HarmonIT project. It is the first in the OpenMI report series, which specifies the OpenMI interface standard, provides guidelines on its use and describes facilities for migrating, setting up and running linked models. Other titles in the series include:

A. **Scope** (this document)

B. Guidelines

C. org.OpenMI.Standard interface specification

D. org.OpenMI.Backbone technical documentation

E. org.OpenMI Development Support technical documentation

F. org.OpenMI.Utilities technical documentation

G. org.OpenMI.Tools technical documentation

H. Additional designs for OpenMI tools

Disclaimer

The information in this document is made available on the condition that the user accepts responsibility for checking that it is correct and that it is fit for the purpose to which it is applied.

The HarmonIT partners will not accept any responsibility for damage arising from actions based upon the information in this document.

Further information

Further information on the HarmonIT project can be found on the project website, http://www.HarmonIT.org.

Further information on the Open Modelling Interface and Environment (the OpenMI) can be found on http://www.OpenMI.org.

Contents

1 Introduction

The OpenMI standard defines an interface that allows time-dependent models to exchange data at run-time. When the standard is implemented, existing models can be run in parallel and share information at each timestep. This is the key to making model integration feasible at the operational level. Model integration helps the understanding and prediction of process interactions and is an essential capability for the achievement of the integrated approach to water management called for in the Water Framework Directive.

1.1 The need for the OpenMI

The *Water Framework Directive* (WFD) calls for *integrated water management* to be put into practice and identifies *whole catchment modelling* as a key part of integrated management. The challenge that this presents is not only that individual catchment processes be modelled but also their interactions. Constructing a single model of all catchment processes is not a feasible option, does not make good use of existing models and doesn't provide the flexibility to try alternative models of individual processes. The only realistic mechanism for whole catchment modelling is *integrated modelling*. This approach links models of different processes and hence allows process interactions to be simulated.

Until now, no generic operational linking mechanism has been developed. However, technological advances in computing, the impetus provided by the Water Framework Directive, co-funding from the European Commission and support from the leaders of earlier attempts has now enabled the FP5 project *HarmonIT* to develop the *Open Modelling Interface and Environment* (the *OpenMI*). The *OpenMI Interface* is a standard interface that enables OpenMI components to exchange data as they run. The *OpenMI Environment* comprises a set of software tools. They facilitate making new and existing model codes *OpenMI-compliant* and they offer facilities to combine OpenMI-compliant components into integrated modelling systems and then run them.

A linkage mechanism, such as the OpenMI, is the key to moving single domain modelling to integrated modelling and integrated modelling from a research exercise to an operational task. It will allow for integrated water management to be put into effect and, hence, the objectives of the WFD to be achieved. After four years development work, testing and review, the OpenMI is now widely accepted as the way forward. Evidence for this view can be found in the number of FP6 proposals and projects that intend to use the OpenMI and in the multinational list of modellers who have attended the first three OpenMI training sessions at their own expense. Hence, the developers believe they have created a software architecture that, by its high quality, will become the European and global standard for linking models from the water domain and, later, any domain. However, adopting the OpenMI requires model developers to make a commitment. Most organizations cannot afford to make that commitment until the OpenMI is widely available in a number of implementations and is properly supported – in other words it becomes a well-maintained standard. Therefore, an OpenMI Consortium is being created to support the user community and sustain the OpenMI into the future.

This document provides an overview of the OpenMI and the consortium.

1.2 Aims and objectives

The aim of the OpenMI is to provide a mechanism by which physical and socio-economic process models can be linked to each other, to other data sources and to a variety of tools at run-time, hence enabling process interactions to be better modelled.

Specific objectives are that the mechanism's design should:

- Be applicable to new and existing models
- Impose as few restrictions as possible on the modeller's freedom
- Be applicable to most, if not all, time-based simulation techniques
- Require the minimum of change to the program code of existing applications
- Keep the cost, skill and time required to migrate an existing model to a minimum so that these factors are not a deterrent to the OpenMI's use
- Be easy to use
- Not unreasonably degrade performance

1.3 Why should organizations adopt the OpenMI?

The discussion above has explained the need for the OpenMI created by the adoption of the WFD. What benefits does it bring to the designated authorities, basin managers, regulators, consultants, modellers and model developers responsible for implementing the WFD? Some of the arguments for adopting the OpenMI put forward by organizations that have already adopted or are considering adopting the OpenMI are:

- Protection and enhancement of existing investment in model development (i.e. it is not necessary to rewrite them completely in order for them to become OpenMI-compliant)
- The simplification of the model-linking process, leading to an improved ability to model process interactions
- The ability to use appropriate model combinations and to swap between different models of the same process, assisting sensitivity analyses and benchmarking
- A reduction in development time and hence cost for decision support systems
- An increased choice for model users, in that they will be able to 'mix and match' models from different sources
- Increased opportunities for model developers in that individual models become more saleable because they can be linked to established systems, enhancing the value of both
- Increased opportunities for the creation of Small and Medium Enterprises (SME), especially from the academic sector

- Increased opportunities to contribute to the implementation and evolution of EU policies

- The opportunity for model developers to concentrate their core business (e.g. computational cores) because they will be able to buy in OpenMI-compliant tools such as GUIs and post-processing tools

- The OpenMI Environment tools for migrating and linking models and monitoring linked model runs (which are available free under an Open Source licence and would otherwise have to be written by the developer)

- The small cost of conversion compared with the cost of writing a whole catchment model from scratch or redeveloping existing models

- The ability for model users to run third-party computational cores in their own environments

- No need to understand other organizations' I/O procedures

- The ability to change a model's code without affecting the linking process or interface

1.4 The development consortium

The OpenMI has been developed by a team drawn from 14 organizations and seven countries co-funded through the European Commission's Fifth Framework programme under contract number EVK1-CT-2002-00090 (the HarmonIT project). Led by the Centre for Ecology and Hydrology, the team comprised the Institute for Inland Water Management and Waste Water Treatment RIZA, DHI Water and Environment, WL Delft Hydraulics, HR Wallingford Group, Universitat Dortmund, Instituto di Ricerca Sulle Acque, the National Technical University of Athens, WRc plc, DHI Hydroinform a.s., Povodi Labe s.p., Hydroprojekt a.s., Alterra B.V. and the Centre National du Machinisme Agricole, du Genie Rural, des Eaux et des Forets.

Design and development has been led by the three major commercial model developers, DHI Water and Environment, Delft Hydraulics and HR Wallingford. The role of the other organizations has been to manage the project, to support the design and development and to test the standard and environment rigorously.

To ensure that the work meets the standards required by the Commission and the scientific and user communities, a panel of experts comprising leading scientists from around the world has reviewed all key documents and advised the Steering Committee. The project's quality assurance plan established procedures for the critical areas of work and covered document and code version control.

2 Overview of the OpenMI

This section describes the main requirements of the OpenMI, the scenarios against which the standard was tested and the functions of the OpenMI interface.

2.1 Requirements

To be useful, the OpenMI must be able to link any models whose interactions need to be simulated in carrying out the requirements of the Water Framework Directive. Therefore, the key requirements of the OpenMI are to:

- Link models from different domains (hydraulics, hydrology, ecology, water quality, economics etc.) and environments (atmospheric, freshwater, marine, terrestrial, urban, rural etc.)
- Link models based on different modelling concepts (deterministic, stochastic etc.)
- Link models of different dimensionality (0, 1, 2, 3D)
- Link models working at different scales (e.g. a regional climate model to a catchment runoff model)
- Link models operating at different temporal resolutions (e.g. hourly to monthly or even annual)
- Link models operating with different spatial representations (e.g. networks, grids, polygons)
- Link models using different projections, units and categorizations
- Link models to other data sources (e.g. databases, user interfaces, instruments)
- Link new and existing (legacy) models with the minimum of re-engineering and without requiring unreasonably high level IT skills
- Not impair performance, especially of large models
- Be based on proven and available technologies (and, in particular, the architecture must be component-based and multi-layered)
- Link models running on different platforms (e.g. Windows, Unix and Linux)
- Be ‘open’ (the interface specification should be placed in the public domain)
- Allow components to be developed using at least the following programming languages: C/C++, C#, Fortran, Delphi/Pascal, Java and Visual Basic.

The remainder of this section shows how these requirements have been met.

2.2 Use cases

To check that the requirements were correctly expressed and to ease the development of an architecture for the OpenMI, a range of scenarios or 'use cases' were identified. Some examples from the full list of cases are shown below:

- Connect two 1D hydrodynamic river models.
- Connect a 1D hydrodynamic model with a water quality transport model.
- Connect a 1D river model with a 3D groundwater model.
- Connect a 1D hydrodynamic river model to vegetation and habitat models.
- Connect a 3D coastal model to a 1D river model.
- Connect a 2D polygon-based root zone model to a 3D regular grid groundwater model.
- Calibrate a rainfall runoff model linked to a hydrodynamic sewerage model.
- Model the propagation of uncertainty through a chain of models.
- Use different units of measurement for the data to be exchanged between models.
- Connect to an agent-based model.

2.3 Terminology

A number of terms are used when describing the OpenMI standard.

As shown in Figure 1, the term *model application* encompasses all parts of the modelling system software that is installed on a computer: for example Mike11, PHABSIM and InfoWorks-RS.

Typically, such systems consist of a *user interface* and an *engine*. Usually, the engine is a generic representation of a process and this is where the calculations for simulating or modelling that process take place. The user supplies information through the user interface and this is converted into the input data for the engine.

The data describes a specific scenario in which the process is to be simulated: for example the Rhine during a time of extreme rainfall. The user runs the engine by selecting an option or pressing a button on the user interface. The engine reads the input, performs the calculations and outputs the results to files or displays.

When an *engine* has read its input it becomes a *model*. For example, an engine may represent the generic process of water flowing in an open channel. When it has read in the data describing the channel network of the Rhine, along with any boundary conditions and rainfall data, it becomes a model of the Rhine in the scenario to be simulated.

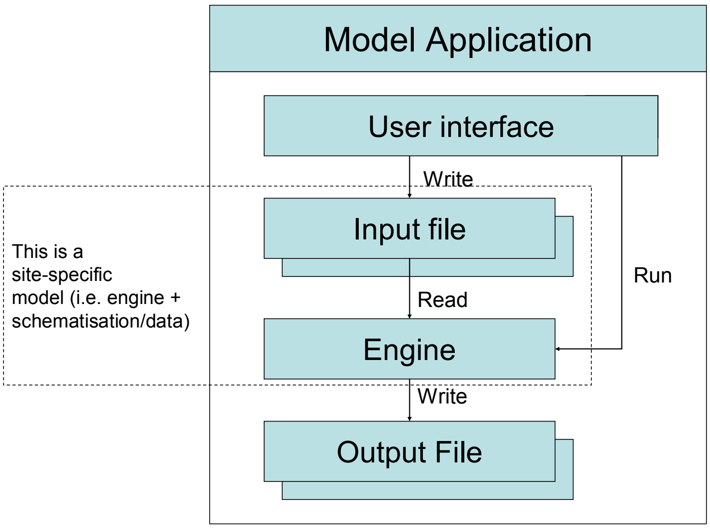

Figure 1 The general structure of a model application

If the code for an engine can be instantiated separately and has a well-defined *interface* through which it can *accept* and *provide* data, then it is an *engine component*. (The engine's *interface* is the part of the code that handles the transfer of data to and from the engine; it should not be confused with the *user interface*, which is the part of the application that the user sees.) The key to enabling models to exchange data lies in standardizing the design of the engine interface. When an engine component implements such a standard interface, it becomes a *linkable component*. An engine that implements the OpenMI interface is called *OpenMI-compliant.*

2.4 The OpenMI standard interface

The OpenMI defines a standard interface that has three functions:

- *Model definition*: To allow other linkable components to find out what items this model can exchange in terms of quantities simulated and the locations at which the quantities are simulated.

- *Configuration*: To define what will be exchanged when two models have been linked for a specific purpose.

- *Run-time operation*: To enable the model to accept or provide data at run time.

Figure 2 shows two model applications whose engines have been made OpenMI-compliant. Their overall structure remains unchanged but each engine is now a component with an OpenMI interface and one component can now get values from another.

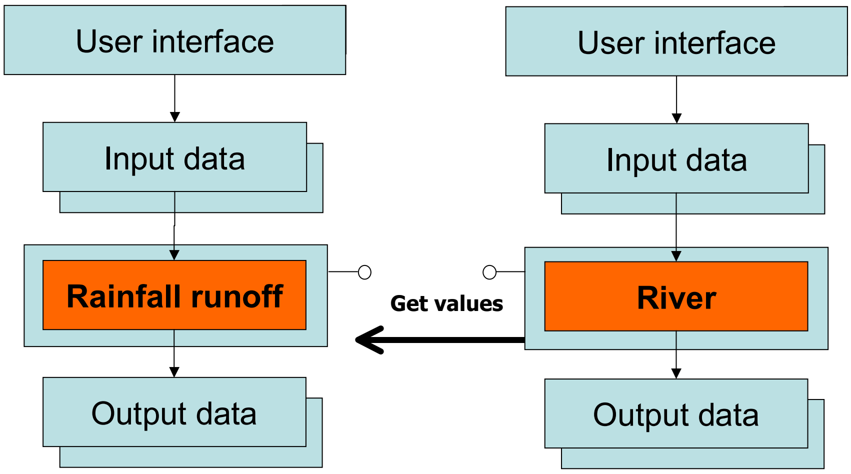

Figure 2 Two applications after migration to the OpenMI standard

Figure 3 illustrates some of the information held in the model definition about the quantities that two models can either accept or provide. The arrow represents a link between the two models and indicates that, in this particular case, runoff produced by the Rainfall Runoff Model will be used to represent lateral inflow in the River Model. There is no requirement to harmonize the terminology; the linking process creates the appropriate cross-reference table.

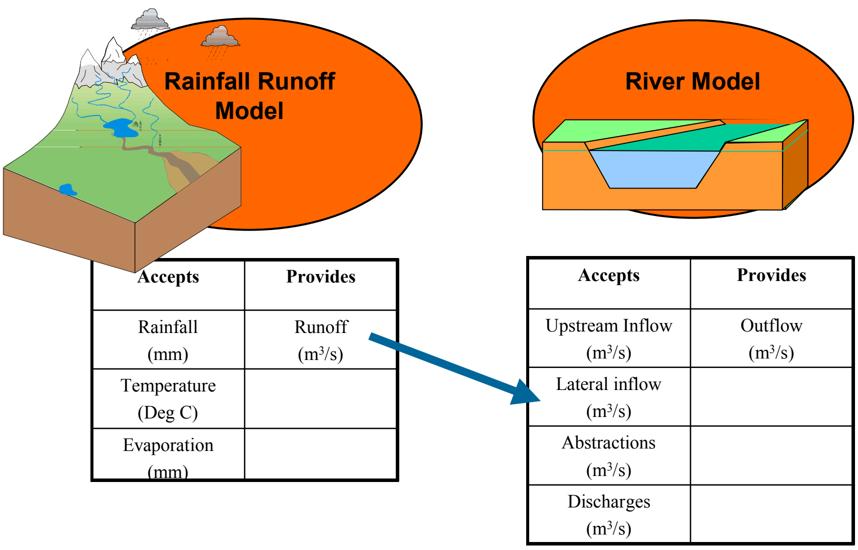

Figure 3 Showing and linking quantities

Figure 4 shows the geographical matching of elements in a river model to those in a groundwater model. The river model is a vector model and each element represents a single stretch; the groundwater model is grid-based, each node being an element. Therefore, in order to link the two models, each element in the river model will usually be linked to several elements in the groundwater model. In any non-trivial situation, this will require the matching of thousands of elements and therefore the process is automated.

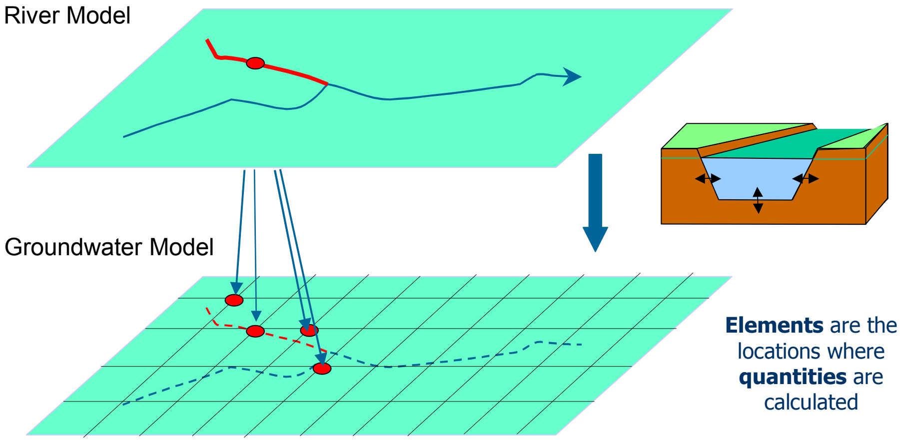

Figure 4 Linking element sets

2.5 An interface-based open standard

The OpenMI provides an intelligent mechanism whereby models running in parallel can exchange data time step by time step. It thus enables process interaction to be represented more accurately than is possible by sequential linkage. It is important to explain that the OpenMI is *neither* a common data-model specification *nor* is it an integrated modelling system.

The OpenMI is 'interface-based':

- Its 'standardized' part is defined as a software interface specification.
- This interface acts as a 'contract' between software components.
- The interface is not limited to specific technology platforms or implementations.
- By implementing this interface a component becomes an OpenMI-compliant component.

The OpenMI is 'open':

- Its specification is publicly available via the Internet (www.OpenMI.org).
- It enables linkages between different kinds of models, different disciplines and different domains.
- It offers a complete metadata structure to describe the numerical data that can be exchanged in terms of semantics, units, dimensions, spatial and temporal representation and data operations.
- It provides a means to define exactly what is linked, how and when.

- Its default implementation and software utilities are available under an open source software license.

The OpenMI is a ‘standard’:

- It standardizes the way data transfer is specified and executed.

- It allows any model to talk to any other model (e.g. from a different developer) without the need for co-operation between model developers or close communication between integrators and model developers.

- Its generic nature does not limit itself to a specific domain in the water discipline or even in the environmental discipline.

Note that the OpenMI enables validation by dimension checks on the quantities linked. However, the OpenMI cannot guarantee that the representation of the process in the component or the link to another component is scientifically valid. That is the responsibility of the modeller, model integrator and user as it should be.

3 Products

A range of products has been delivered by the HarmonIT project. The partners in the HarmonIT project also intend to establish an OpenMI consortium that will continue to deliver OpenMI products and services.

3.1 Products delivered by the HarmonIT project

The HarmonIT project has delivered a number of products.

3.1.1 The OpenMI standard interface specification

The OpenMI standard interface specification defines the interface that an engine component must provide for it to be OpenMI-compliant. This has been placed in the public domain and may be found on the OpenMI website at www.OpenMI.org.

3.1.2 The OpenMI environment

The OpenMI environment is software that assists in the implementation of the standard. The OpenMI 2005 software release includes the compiled .NET assemblies, the source code of all packages and their documentation. The software is released under Lesser GPL license conditions and is available on www.OpenMI.org and www.SourceForge.net/projects/OpenMI.

3.1.3 The OpenMI documentation

All aspects of the project are documented in the OpenMI report series, comprising the following titles:

A. Scope (this document)
Describes the scope of the OpenMI architecture and the organization behind it.

B. Guidelines
Describes how to migrate, link and run OpenMI-compliant models. The document includes sample code and tutorial examples.

C. org.OpenMI.Standard interface specification
Describes the interface specification of the org.OpenMI.Standard namespace (the OpenMI standard interface specification). The specification is expressed in Universal Modelling Language (UML) and in API terms. This specification has to be adopted for a model to become OpenMI-compliant.

D. org.OpenMI.Backbone technical documentation
Describes the default implementation (i.e. the classes that implement the org.OpenMI.Standard interface). This implementation is the basis of the OpenMI environment.

E. org.OpenMI.Development Support technical documentation

Describes a generic set of low-level classes that can be used in the development of an OpenMI system.

F. org.OpenMI.Utilities technical documentation

Describes the org.OpenMI.Utilities namespace, which contains useful low-level classes and packages that have been tailored to help with the development of OpenMI-compliant systems. Their use is entirely optional and is not a requirement of compliance. Typically, the classes within this namespace have been developed to reduce the amount of re-engineering needed when migrating existing model engines and software systems to become OpenMI-compliant. In particular, the utilities provide facilities to support the wrapping of legacy code. The namespace contains packages for wrapping, buffering, spatial mapping, advanced control features, configuration and deployment.

G. org.OpenMI.Tools technical documentation

Describes the org.OpenMI.Tools namespace, which contains the front-end tools to interact with OpenMI-compliant components and configure system combinations.

H. Additional designs for OpenMI tools

Describes the part of the org.OpenMI.Tools namespace containing user interface components and other classes that can interact with the OpenMI software packages at various levels.

3.2 Future products and services

The HarmonIT project has delivered a technology that has the potential to become an international standard for model linkage and data exchange. However, this potential will only become a reality if the OpenMI is used in practice by a wide user community.

The partners in the HarmonIT project therefore intend to establish an OpenMI consortium, whose mission will be:

- To provide global access to the OpenMI standard and associated software
- To promote the wide uptake and use of the OpenMI
- To deliver ongoing enhancements to the OpenMI as required by its users
- To improve the OpenMI products

The consortium will be a self-sustaining organization that can ensure the future maintenance of the OpenMI as a worldwide-applied software standard for model linkage in environmental and other domains.