The OpenMI interface
for flexible and dynamic coupling

Workshop on
"Coupling Technologies for Earth System Modelling : Today and Tomorrow"
CERFACS, Toulouse (France) - Dec 15th to 17th 2010

Stef Hummel

December 15th 2010
Outline

- Introduction
- OpenMI concepts
- OpenMI 2.0 versus 1.4
- GUI’s
- Interfaces / Model Wrappers
- Applications, developments, etc.
Technological breakthroughs and innovation are essential to maintain the habitability of coastal and fluvial areas. Through an integrated approach, Deltares develops knowhow for innovative solutions, to make living in deltas safe, clean and sustainable, worldwide.
Deltares expertise

- Urban land and water management
- Integrated coastal management
- Integrated water management
- Climate
- Safety and risks
- Operational forecasting systems & management systems
- Subsurface and water quality
- Spatial planning and ecology
- Linear infrastructure
- Innovation management
- Strategic analysis
- Hydraulics
- Offshore
- Intake & outfall systems
- Dikes
Series of EU-projects:
- **HarmoniCA** (Concerted Action – good modeling practice)
- **HarmoniRiB** (River Basins – uncertainty handling)
- **HarmoniIT** (Information technology):
  - Develop a Standard for model integration
  - Provide a reference implementation of that standard
  - Provide utilities for letting existing models adhere to that standard

Result of HarmoniIT:
- **OpenMI** 1.0 – 1.4

Participants:
- Developers: Deltares (Delft Hydraulics), DHI, Wallingford Software
- Testers/users: Many other institutes and companies …
Application of OpenMI in:

- Various projects at the involved partners
- SeamLess (syst. for agric. modeling, linking europ. sci. and soc.)
  - Agricultural application (GIS-environment, different type of models)
  - Java environment
- OpenMI-Life ("bringing OpenMI to life")
  - Specific focus: extend the standard to
    - other domains than the water related one
    - other type of components that numerical engines
  - Apply OpenMI to some specific use cases
- Result of OpenMI-Life:
  - OpenMI 2.0 (standard, limited reference implementation)
OpenMI history (3)

Participants in various stages and roles

Halcrow, UNESCO-IHE
RWTH-Aachen
OpenMI purpose (1)
Provide a *minimal and very general* interface data exchange between models

- **What** is exchanged? (Quantity or Quality)
- **Where** is it exchanged? (ID-based or Geo-referenced set of locations)
- **When** is it exchanged? 'TimeSet': specification of time stamp(s) or time span(s)
- **How** is it transformed if the provided data does not fit the required ones

A *minimal and very general* interface:

- to impose as little requirements on a model component as possible
- to make sure that existing specifications can be wrapped in the OpenMI specifications
Concepts: Linkable Component / Exchange Item

Linkable Component

Output Items

Provider / Consumer relation

Adapted-Output

Input Items

Linkable Component
The Input item specifies its needs: what, where, when, and then performs a GetValues-call to the providing Output or Adapted Output.

Quantity and ElementSet are usually static, but may vary.

IInput

- SetValues(newValues)

I(Adapted)Output

- GetValues(specified Input)

(input.Quantity = ... )
(input.ElementSet = ...)
input.TimeSet = { someTimeSpan }
valueSet =
(adapted)output.GetValues(inputItem)
Concepts: Adapted Outputs

Component 1
- Output 1
- Output 2
- Output 3
  - Spatial adaptation A
  - Spatial adaptation B
  - Time interpolation
    - SI-conv.

Component 2
- Input 1
- Input 2
- Input a

Component 3
- Input b
- Input c

SI-conv.
<table>
<thead>
<tr>
<th>Concepts: Quantity/Quality</th>
</tr>
</thead>
</table>

### Quantity

- **Caption**: “Runoff”
- **Description**: (optional additional explanatory description)
- **Value Type**: (double, integer, etc.)
- **Unit**:
  - **Caption**: “CFS”
  - **Description**: “Cubic feet per second”
  - **ConversionFactorToSI**: 0,0283168439
  - **OffsetToSI**: 0
  - **Dimension**: (e.g. \( L^3 T^{-1} \))

### Quality

- **Caption**: “soil type”
- **Description**: (optional additional explanatory description)
- **Categories**:
  - **Caption**: “sand 1”
  - **Description**: “course sand”
- **IsOrdered**
Concepts: ElementSets

- Rainfall, IsoHyet 1
- Rainfall, IsoHyet 2
- SOBEK RR Catchments
- MOUSE Sewer
- Ground-Water

ISIS River

23 December 2010
OpenMI 2.0 vs 1.4: Other type of models

- Time progressing numerical Model
- Constant Value Provider
- 0-Dimension, Time Series
- GIS
- Optimizers, Calibration tools, Scenario Managers
- Analytic Function $f(x,y,z,t)$
- Data Base or File Reader
OpenMI 2.0 vs 1.4: Flexible Linking

Instead of creating a link for each connection, where the link does all transformations, use the required adapted outputs.
Useful for tools that (repeatedly) analyse certain "what if" scenario's:

**Calibration:**
*What if* the bottom friction is adjusted, will the model perform better?

**Decision support tools:**
*What if* somewhere upstream more water is extracted, will there be problems downstream?

Set the values into the input item, according to its location specification and/or its time specification.
GUI’s (1)
GUI’s (3)
OpenMI interfaces: ILinkable Component

**ILinkableComponent**
- **ILinkableComponentStatus**
  - Enum
  - Created
  - Initializing
  - Initialized
  - Validating
  - Valid
  - WaitingForData
  - Invalid
  - Preparing
  - Updating
  - Updated
  - Done
  - Finishing
  - Finished
  - Failed

- **ILinkableComponentStatusChangeEventArg**
  - Class
  - Event

- **ILinkableComponent**
  - Interface
  - **ILinkable**
  - Properties
    - **AdaptedOutputFactor**
      - Arguments: `get`: IL...
      - Properties: `get`: IL...
      - Events: `StatusChanged`: Event...

- **IManageStateExtension**
  - Interface
  - Methods
    - `ClearState(IIdentifiable stateId)`: void
    - `KeepCurrentState()`: IIdentifiable
    - `RestoreState(IIdentifiable stateId)`: ...

- **IByteStateConverter**
  - Interface
  - Methods
    - `ConvertFromString[Byte](Byte[] byteArray)`
    - `ConvertToByteArray(IIdentifiable stateId)`

- **ITimeExtension**
  - Interface
  - Properties
    - `TimeExtent {get; set}`: ITimeSet

- **ITimeSpaceComponent**
  - Interface
  - **IEaseLinkableComponent**
  - **ITimeExtension**
OpenMI interfaces: IInput / IOutput
Wrapping Legacy Code (1)

```
do t = 1, nT
  t == 1? Initialize
  Initialize TimeStep
  Compute TimeStep
  Finalize TimeStep
  t == nT? Finalize
  Finalize
```

---

23 december 2010
Wrapping Legacy Code (2)

computational kernel

- Initialize
- PerformTimeStep
- Finalize

Set<varType>(varId)
Get<varType>(varId)
SetValues(…)
GetValues(…)

23 december 2010
OpenMI Applications

- Dutch National Hydrological Instrumentary
  - Drought studies
- 1D Open Water: Holland in “Subdomains”
- 1D Open Water / Industrial model
  - Pump Station Design
- Generic Input for 3-D Open Water
- Waves / 2-D Open Water
- 1D / 2D Open Water
- Wallingford’s FluidEarth
- ....
OpenMI Implementations / Environments

OpenMI 1.4:
- OATC’s *) System Development Kit (C#, Windows/.NET)
- OATC’s SDK (C#, Linux/Mono)
- Alterra’s SDK for SeamLess (Java)

OpenMI 2.0 developments:
- OATC’s limited SDK 2.0 (C#, Windows/.NET)
- OATC’s limited SDK 2.0 (C#, Linux/Mono)
- SDK for java (initiated by Alterra)
- Fluid Earth SDK, Pipistrelle GUI (HR Wallingford)
- HydroDeskTop (CUAHSI)

*) OpenMI Association Technical Committee
OpenMI Co-operations

- EPA FRAMES
- ESMF webservice
- CSDMS (Community of Surface Dynamics Modeling System)

Modeling Tool
- OpenMI based linking implemented on top of the Common Component Architecture (CCA)
OpenMI sites

- www.openmi.org
- wiki.openmi.org
- http://sourceforge.net/projects/openmi