Integrated Modeling Overview: OpenMI and EMIT

Anthony Castronova
Utah State University
OpenMI Overview and Applications
Model Integration
Tight Integration

- Bound by source code
- Enables complete control over integrated modeling system
- Can be optimized for performance
- Requires a great deal of expertise
- Source code must be available
- Requires customized integration for every model system
Loose Integration
Open Modeling Interface

- Software Interfaces for Component Modeling
- EU Water Framework Directive
- C#, Java
- DHI, Delft (Deltares), Wallingford Software (Innovyze)

Concepts
- Timestep data transfer (pull-based)
  - Ask for what you want, get what you asked for
- Data Representation
  - Exchange Items (what is it, where is it)
  - Variable, Unit, Element Set (geometric data)

Fig. 2. Two applications after migration to the OpenMI standard.
OpenMI Timeline

- **2007 OpenMI 1.4**
  - C#, Java standard interfaces
  - C# development API

- **2010 OpenMI 2.0**
  - C#, Java standard interfaces
  - C# development API
  - more flexible component linking and control flow, enhanced data values support

- **2011/2012 FluidEarth**
  - Graphical user interface for building coupled models

- **Software developers from many difference commercial modeling companies**
  - Professional software development
  - Limited time
The OpenMI Standard Concepts

- Ground Water
- River
- Rainfall Runoff

<table>
<thead>
<tr>
<th>Accepts</th>
<th>Provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rainfall (mm)</td>
<td>Runoff (m³/s)</td>
</tr>
<tr>
<td>Temperature (Deg C)</td>
<td></td>
</tr>
<tr>
<td>Evaporation (mm)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accepts</th>
<th>Provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upstream Inflow (m³/s)</td>
<td>Outflow (m³/s)</td>
</tr>
<tr>
<td>Lateral inflow (m³/s)</td>
<td></td>
</tr>
<tr>
<td>Abstractions (m³/s)</td>
<td></td>
</tr>
<tr>
<td>Discharges (m³/s)</td>
<td></td>
</tr>
</tbody>
</table>
Simulation Control

Linear chain (unidirectional)

A requests B, B requests C, C requests D.
D does its work and returns data to C, C does its work and returns data to B, etc.

Request for data
What
Where
When

Linear chain (bidirectional)

A requests B, B requests C, C requests B
B returns a best guess to C. C does its work and returns data to B. B does its work and returns data to A.

Reply with data
Advantages

• **Collaboration and cross-disciplinary studies**
  - Large problems that can not be explored by stand alone models.
  - Predict the wider implications of policies and projects

• **Modular development**
  - Simplifies model substitution
  - Helpful when changing spatial scales, and performing uncertainty or sensitivity analyses
  - Reduces Model creep

• **Model and source reuse**
  - Eliminate the need to redevelop models
  - Cross-disciplinary model development

• **Community Adoption**
  - Simple Model Wrapper
  - Simple Script Wrapper
  - PyOpenMI
  - HydroCouple
EX1: Integrate OpenMI with Existing Architecture

1. Establish a method for mapping data between the OpenMI and HIS
   1. Conceptual mapping
   2. Metadata mapping
   3. Observations vs Model computations

2. Build integrated modeling environment
   1. Leverage data mapping
   2. Build from of the OpenMI standard
   3. Dedicated components for reading and writing data to and from ODM databases

3. Construct some models to leverage new tech.
   1. Urban Watershed configuration
   2. ET configuration (dominated by observations)
   3. Rural watershed configuration
CUAHSI HIS

- Consortium of Universities for the Advancement of Hydrologic Science, Inc. Hydrologic Information System

- Collaborative effort
  - University of Texas, Austin
  - Utah State University
  - San Diego Super Computing Center
  - Idaho State University
  - City College of New York
  - University of South Carolina

- NSF Funded ($4.7 Million 2007-2011)

- Goals
  - Make the nation's water information universally accessible and useful
  - Provide access to data sources, tools and models
  - Enable the synthesis, visualization and evaluation of the behavior of hydrologic systems.
Hydrologic Information System (HIS)

- Web service network of hydrologic resources
- Enables users to publish data
- Searching across distributed HIS data

- Opensource GIS
- Entry point to the HIS observation data
- Extendable through plugin development
Observations Data Model (1.1)
Data Mapping

- **Hydrologic Information System**
  - Formal representation of observations data
  - Comprehensive metadata
  - Considers only point geometries

- **Open Modeling Interface**
  - Standard for describing model data
  - Transferring and sharing of data
  - More general
HydroModeler

- Component modeling plugin for the HydroDesktop
- Extends the original OpenMI configuration editor (OmiED)
- Provides a more extensible modeling environment
- Built on OpenMI 1.4

http://his.cuahsi.org/hdhelp/extensions/hydromodeler/HydroModeler.html
Model Configuration for ET

1. Db Reader
   - Transfer of dew point, maximum, minimum, and monthly averaged range temperatures

2. Db Reader
   - Transfer of wind speed, and dew point, maximum, minimum, and averaged temperatures

3. Solar Radiation
   - Calculate solar radiation

4. Db Writer
   - Write Etsz values to local database repository
   - Transfer Etsz

5. ASCE ETsz
   - Calculate Etsz

Increment time

Increment Trigger
ET Results

Diagram showing the process of calculating ET Results with steps:
1. Calculate ETsz from Equation 1
2. Transfer solar radiation
3. Transfer of dew point, maximum, minimum, and monthly averaged range temperatures
4. Calculate solar radiation using Equations 2 through 7
5. Transfer ETsz
6. Write ETsz values to local database repository
7. Increment time

Graph showing PET - millimeters per day with data points for Asheville, NC and Greenville, SC.
Coweeta Watershed

- TOPModel
- Located in western NC
- Long Term Ecological Research
- Watershed 18, 0.12 km²
- Readily available observations
Modeling Web using REST/OGC Web Processing Service

- Models exposed as services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard.
- Get Capabilities gives metadata about models on server
- Describe Process gives metadata about a specific model service
- Execute is more complex than a typical WPS to support model time stepping (e.g., session state with users must be maintained)
Simulation and Results

- **DbReader**
  - Supplies CUAHSI HIS observation data

- **Hargreaves**
  - Calculates PET using temperature data.

- **Wps:Topmodel**
  - Wraps the web TOPMODEL web resource into an OpenMI-compliant form

---

**Diagram:**

- **Db Reader**
  - Transfer precipitation
  - Transfer minimum, maximum, and average temperature

- **Hargreaves PET**
  - Calculate potential evapotranspiration

- **Trigger**

- **Query WPS using URI**

- **GET**
  - Send time-dependent input data
  - Save time-independent data

- **Web Resource**
  - Perform Computation

- **Data**
  - Send time-independent data
EX3: Urban Infrastructure

- Downtown Columbia floods often
  - Property damage
  - Infrastructure damage

- Short, high intensity storms
  - 2.26 in 1 hour
  - River depth from 10in to 10.74ft in 1 hour
Model Design

PDF of Young’s Modulus

PDF of Beam Support Displacement at h = 7.5 ft

CDF of Bridge Failure
Simulation Results
Lessons Learned

- OpenMI was used to provide runtime interaction between OpenMI and HIS
  - DbReader can seamlessly extract observation data during a simulation
  - DbWriter automatically writes model data back to the HIS repository

- This approach was successfully extended using web services
  - OpenMI concepts are transferrable beyond their original use cases
  - Provide acceptable description of model input and outputs.

Platform dependence
- Becoming more important as more scientists are using Linux and OSX.
- C# is not a scientist friendly language (high learning curve, not taught to civil graduate students)

OpenMI core development efforts
- Limited development time results in lengthy gaps between releases
- Releases 1.4 and 2.0 have significant differences
- Lack of sufficient GUI (OmiED, Pipistrelle, FluidEarth)
iUtah Coupled Modeling
iUTAH: innovative Urban Transitions and Aridregion Hydro-sustainability

- Statewide effort
- $20 million competitive award from NSF EPSCoR
- Research capacity building
- Interdisciplinary and multi-institution
- Focused on sustainable management of Utah’s water resources

Courtesy of Amber Spackman-Jones
iUTAH Research Focus Areas

1. **Eco-hydrology**
   Expand Utah’s capacity in the natural sciences through instrumentation of 3 watersheds.

2. **Social and Engineered Water Systems**
   Studying demographic characteristics, water use behaviors, water infrastructure, and other measures of urban form.

1. **Interdisciplinary Modeling and Visualization**
   Development of interdisciplinary models of socio-eco-hydrological systems to determine how changes in water availability and use alter water quantity and quality.
Ecohydrologic observatory deployed in 3 watersheds in northern Utah, USA: Logan River, Red Butte Creek, Provo River.

Watersheds with similar water source (high elevation snow) but different land use transitions.

Measures aspects of water inputs and outputs and water quality over mountain-to-urban gradient.

Mix of aquatic and terrestrial in situ and re-locatable sensors.

Courtesy of Amber Spackman-Jones
**Gradients Along Mountain to Urban Transitions (GAMUT) Network**

### Climate/Terrestrial Sites

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Instrument</th>
<th>Variables Measured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campbell</td>
<td>HC253</td>
<td>Air Temperature and Relative Humidity</td>
</tr>
<tr>
<td>Apogee</td>
<td>ST110</td>
<td>Air Temperature</td>
</tr>
<tr>
<td>Campbell</td>
<td>CS106</td>
<td>Barometric Pressure</td>
</tr>
<tr>
<td>RM Young</td>
<td>5303</td>
<td>Wind Speed/Direction</td>
</tr>
<tr>
<td>Geonor</td>
<td>TB-200</td>
<td>Precipitation</td>
</tr>
<tr>
<td>Judd</td>
<td>Depth Sensor</td>
<td>Snow Depth</td>
</tr>
<tr>
<td>Hukseflux</td>
<td>NR01</td>
<td>Incoming and Outgoing Shortwave and Longwave Radiation</td>
</tr>
<tr>
<td>Apogee</td>
<td>SP-230</td>
<td>Incoming Shortwave Radiation</td>
</tr>
<tr>
<td>Apogee</td>
<td>SQ-110</td>
<td>Incoming and Outgoing Photosynthetically Active Radiation</td>
</tr>
<tr>
<td>Apogee</td>
<td>Si-111</td>
<td>Surface Temperature</td>
</tr>
<tr>
<td>Acclima</td>
<td>ACC-SEN-SDI</td>
<td>Soil Moisture, Temperature, and Conductivity at 5 cm, 10 cm, 20 cm, 50 cm, 100 cm below ground</td>
</tr>
<tr>
<td>Campbell</td>
<td>CS210</td>
<td>Enclosure Humidity</td>
</tr>
<tr>
<td>Campbell</td>
<td>18166</td>
<td>Enclosure open door sensor</td>
</tr>
</tbody>
</table>

### Aquatic Sites

- Pressure Transducer housing
- Sonde housing
- Stage plate
- Turbidity Sensor housing
The Data Deluge

One day = 96 observations
One week = 672 observations
One month = 2880 observations
One year = 35,040 observations
So far (~1.5 years) = 55,000+ observations

Times 14 Aquatic Sites with ~26 Variables
Times 14 Climate Sites with ~74 Variables
Plus different versions of the data = 43,400,000+ observations

We Want To Use This Data!

Courtesy of Amber Spackman-Jones
Data Movement and Storage
Framework Design Considerations

- Diverse scientific needs
  - Hydrology, climate science, agriculture, city planning, green infrastructure, etc.

- Community engagement
  - Needs to be easy to implement
  - Supports various platforms

- Data ingestion
  - Massive amount of data
  - Seamless transfer of data into models
  - **Data Centric**
Data Centricity

Logan ODM Database

Red Butte ODM Database

Provo ODM Database

ODM Version 2
Conceptual Model

- Components
  - Source or executable wrapped with standard interfaces

- Configurations
  - Linked arrangement of components.
  - Links defining the transfer of data

- Simulations
  - Executed configurations and their results

- Coordination
  - Central data coordinator
  - Execution of models and transfer of data

Storm Water Management Model
Conceptual Model

- **Components**
  - Source or executable wrapped with standard interfaces

- **Configurations**
  - Linked arrangement of components.
  - Links defining the transfer of data

- **Simulations**
  - Executed configurations and their results

- **Coordination**
  - Central data coordinator
  - Execution of models and transfer of data
Conceptual Model

- **Components**
  - Source or executable wrapped with standard interfaces

- **Configurations**
  - Linked arrangement of components.
  - Links defining the transfer of data

- **Simulations**
  - Executed configurations and their results

- **Coordination**
  - Central data coordinator
  - Execution of models and transfer of data
Conceptual Model

- **Components**
  - Source or executable wrapped with standard interfaces

- **Configurations**
  - Linked arrangement of components.
  - Links defining the transfer of data

- **Simulations**
  - Executed configurations and their results

- **Coordination**
  - Central data coordinator
  - Execution of models and transfer of data
Data Flow Mechanisms

- **Feed Forward**
  - Sequentially executed
  - Data passed in a single direction
  - Data Models, closed source software

- **Time Stepping**
  - Models advancing in time together
  - Resolving boundary conditions

- **Looping (Hybrid)**
  - Pushing loop
  - Data / Feed models will not execute, but still pass data along at requested times

```plaintext
[ 0.14, 0.69, 0.90, ...]
[ 0.49, 0.16, 0.70, ...]
[ 0.66, 0.40, 0.53, ...
[...]
[...]

[ 0.55, 0.90, 0.01, ...,]
[ 0.12, 0.60, 0.07, ...]
[ 0.60, 0.04, 0.39, ...
[...]
[...]

[ 1.55, 1.90, 1.01, 1.12,
  1.60, 1.07, 1.60, 1.04,
  1.39, ...
```
Workflow Mechanisms

- **Feed Forward**
  - Sequentially executed
  - Data passed in a single direction
  - Data Models, closed source software

- **Time Stepping**
  - Models advancing in time together
  - Resolving boundary conditions

- **Looping (Hybrid)**
  - Pushing loop
  - Data / Feed models will not execute, but still pass data along at requested times

```plaintext
[ 0.14, 0.69, 0.90, ...]
[ 0.49, 0.16, 0.70, ...]
[ 0.66, 0.40, 0.53, ...]
[...]
[...]
[ 0.55, 0.90, 0.01, ...]
[ 0.12, 0.60, 0.07, ...]
[ 0.60, 0.04, 0.39, ...]
[...]
[...]
[ 1.55, 1.90, 1.01, 1.12, 1.60, 1.07, 1.60, 1.04, 1.39, ...]
```
Workflow Mechanisms

- **Feed Forward**
  - Sequentially executed
  - Data passed in a single direction
  - Data Models, closed source software

- **Time Stepping**
  - Models advancing in time together
  - Resolving boundary conditions

- **Looping (Hybrid)**
  - Pushing loop
  - Data / Feed models will not execute, but still pass data along at requested times
Goals

1. Extend existing coupled modeling architectures to engage a broad audience of modelers.

2. Design software wrappers to facilitate the transfer and ingestion of data between models in a generic manner.

3. Develop code that does not obstruct the functionality of legacy and newly developed software models.

4. Automatic transformation of temporally and spatially misaligned data between coupled models during simulations.

5. Access to environmental data (e.g. GAMUT) and server-side simulation results (e.g. NetCDF).

6. Develop a method to track model data provenance.

7. Maintain platform independent software to appeal to a diverse audience of modelers and models.
Features

• Modeling on an Open Platform
  o FOSS Libraries
  o FOSS Codebase

• Integration of Legacy Code
  o Operate on Windows, Linux, Mac
  o Language Interoperability
  o C, C++, Fortran, Python

• Data Transformations
  o Temporal misalignment
  o Spatial misalignment

• Simulation Workflow
  o Loop Driven (timestepping push control)

• Investigates how data can be passed to and from databases during simulation
  o Remotely stored data
  o Large datasets
  o Long term data archival
  o Complete record of all simulation results
  o Comprehensive data model for simulation archival
Data Storage

**Simulation Information Model**

*Data Storage*

**Data Grouping**

- RelatedDatasets
  - RelatedDatasetID
  - DatasetID
  - RelatedDatasetName
  - RelatedDatasetDescription
  - VersionCode

- DataSets
  - DatasetID
  - DatasetName
  - DatasetDescription
  - DatasetCode
  - DatasetAbstract

- DataSetsResults
  - DatasetResultsID
  - DatasetResultsName
  - DatasetResultsDescription
  - DatasetResultsCode
  - DatasetResultsAbstract

**Sampling Features**

- SamplingFeatureID
  - SamplingFeatureName
  - SamplingFeatureDescription
  - SamplingFeatureCode
  - SamplingFeatureAbstract

**Variables**

- VariableID
  - VariableName
  - VariableDescription
  - VariableCode
  - Variable abstract

**Results**

- ResultID
  - ResultName
  - ResultDescription
  - ResultCode
  - ResultAbstract

**Actions**

- ActionID
  - ActionName
  - ActionDescription
  -ActionCode
  - ActionAbstract

**Units**

- UnitID
  - UnitName
  - UnitDescription
  - UnitCode
  - UnitAbstract

**Organizations**

- OrganizationID
  - OrganizationName
  - OrganizationDescription
  - OrganizationCode
  - OrganizationAbstract

**Affiliations**

- AffiliationID
  - AffiliationName
  - AffiliationDescription
  - AffiliationCode
  - AffiliationAbstract

**People**

- PersonID
  - PersonName
  - PersonDescription
  - PersonCode
  - PersonAbstract

**Methods**

- MethodID
  - MethodName
  - MethodDescription
  - MethodCode
  - MethodAbstract

**Time Series Results**

- TimeSeriesResultID
  - TimeSeriesResultName
  - TimeSeriesResultDescription
  - TimeSeriesResultCode
  - TimeSeriesResultAbstract

Coupling: a simulation can have multiple related simulations that are defined by a relationship type. This is used to "link" simulations into coupled workflows.

Ownership: models and simulations are described using the organization, affiliation, and people entities. This enables ownership to be archived which is essential when publishing study results.

Implementation: Every simulation consists of sampling features which define the geometry of a result. PostgreSQL and PostGIS allow simulation data to be queried based on location, as well as enable transformations using built-in geographic functions.

Reusability: Each simulation is related to input and output data, which can be used to recreate results and share simulations.

Specific result types are encoded by the ODM2 results extension. They can be characterized as time series, profiles, measurements, transactions, etc.
Where are we going?

- **Next few months**
  - All GAMUT data will be made available for use in models
  - Results storage in local SQLite database.
  - Complete integration of NetCDF/HD5 datasets

- **Within the next year**
  - **SWMM**, urban watershed model
  - **TOPMODEL**, topographically-based watershed model
  - **TOPKAPI**, fully distributed extension of TOPMODEL
  - **UEB**, Utah Energy Balance Model
  - **WRF-Hydro**, Climate simulation data via THREDDS/OpenDAP
  - **PIHM**, finite volume hydrologic model
  - **WEAP**, Water Evaluation And Planning
  - **SLEUTH**, Land Change Model
  - **NetLogo**, multi-agent programmable modeling environment.
Anthony Castronova

Research Assistant Professor
Utah State University
tony.castronova@usu.edu

https://github.com/castronova