The Object Modeling System
Framework for Water- and Environmental-Model Development and Application for
Resource Management

George H. Leavesley & O. David

Department of Civil and Environmental Engineering, Colorado State University, Fort Collins, CO; U.S. Geological Survey, Denver, CO (retired)
STARTING POINTS

• There are no universal models
• Models for different purposes require different levels of detail and comprehensiveness
• Appropriate model process conceptualizations are a function of problem objectives, data constraints, and spatial and temporal scales of application
• Experimental science builds on hypothesis testing and interpretation based on earlier published hypotheses and results
• Modelers tend to build from the ground up because existing models are not well designed for incremental improvement by others
Object Modeling System (OMS)

Integrated systems of computer software developed to:

• Provide a common framework for model development and operational applications
• Facilitate the integration of multi-disciplinary modeling approaches for use in addressing complex, multi-objective, resource-management problems
• Provide a toolbox-approach to the development of resource-management, decision-support systems
Object Modeling System (OMS)
• Modular Modeling System (MMS)
  • Object Based
  • C and Java Language Framework
  • Fortran, C module support
  • Windows environment

• Object Modeling System (OMS)
  • Object Oriented
  • Java Language Framework
  • Java, C, C++, Fortran module support
  • Platform independent
TOOL BOX MODELING VIEWS

- Research Model Developer
- Application Model Developer
- Model User
- Resource Manager
- Policy Maker
LEVELS OF MODULAR DESIGN

- PROCESS
- MODEL
- FULLY COUPLED MODELS
- LOOSELY COUPLED MODELS
- RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
- ANALYSIS AND SUPPORT TOOLS

Single Purpose

Multi-objective, Complex
LEVELS OF MODULAR DESIGN

• PROCESS
• MODEL
• FULLY COUPLED MODELS
• LOOSELY COUPLED MODELS
• RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
• ANALYSIS AND SUPPORT TOOLS
Module / Component Library

<table>
<thead>
<tr>
<th>Precip</th>
<th>Evap</th>
<th>Solar Rad</th>
<th>Soil</th>
<th>Snow</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td></td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- Red: A
- Blue: B
- Gray: C
- Blue: D
- Red: E
- Blue: F
CRITERIA AND RULES FOR GOOD MODULE DESIGN

Modules should
• relate directly to real world components or processes
• have input and output variables that are measurable values
• communicate solely via these input and output variables

Model Building Tools

Module Locations

Available Modules

totet_jh_prms.f
totet_epan_prms.f
totet_hamon_prms.f
Time – Space – Process Configurations
LEVELS OF MODULAR DESIGN

• PROCESS
• MODEL
• FULLY COUPLED MODELS
• LOOSELY COUPLED MODELS
• RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
• ANALYSIS AND SUPPORT TOOLS
Figure 3. Diagram showing one ground-water cell with stream depicting properties used in calculation of the streambed leakage for a subreach.
GSFLOW -- Coupled PRMS, MODFLOW, SFR, and Unsaturated Zone Models

- PRMS to SFR2
- PRMS to MODFLOW
- MODFLOW to SFR2
- Unsaturated Zone Model:
  - PRMS to UZF
  - UZF to MODFLOW

Streamflow
LEVELS OF MODULAR DESIGN

• PROCESS
• MODEL
• FULLY COUPLED MODELS
• LOOSELY COUPLED MODELS
• RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
• ANALYSIS AND SUPPORT TOOLS
LOOSELEY COUPLED MODELS

- Watershed Model
- Hydraulics Model
- Fish Model

Database

- Modular Model
- Off-the-shelf Model

Data Management Interface (DMI)
LOOSELEY COUPLED MODELS

- Watershed Model
- Hydraulics Model
- Fish Model
- MMS Model
- Off-the-shelf Model
- Database
- Data Management Interface (DMI)
- Model Management Interface (MMI) [XML]
Aquatic Habitat Models

Watershed Model

Hydraulics Model

Fish Model

Depth HSI - Brown Trout

Hydraulic Radius = Area / Wetted Perimeter
Discharge = Cross Section Area * Mean Channel Velocity
Aquatic Habitat Models Results
LEVELS OF MODULAR DESIGN

• PROCESS
• MODEL
• FULLY COUPLED MODELS
• LOOSELY COUPLED MODELS
• RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
• ANALYSIS AND SUPPORT TOOLS
Database-Centered Decision Support System

RiverWare System
River and Reservoir Management Models

Data Management System
Hydrologic Database (HDB)

Data Sources
Real-time Telemetry
SCADA
NEXRAD

Query/Display/Analysis
GIS
Statistics
Trade-Offs
Risk

Modular Modeling System
MMS
Watershed and Ecosystem Models
RiverWare – Lower Colorado River
Upper Gunnison River DSS

Hydromet
Real-time climate data feed

Hydrologic Database

DMI

RiverWare
Reservoir and River System Operations Model

Object User Interface
Interface for data visualization and modeling

Modular Modeling System
Precipitation/Runoff Model (PRMS)
OBJECT USER INTERFACE (OUI)

- Java based, modular interface
- Data retrieval and management
- Visualization of data and model parameters
- Model output analysis and visualization
- Model animation
- Manage single and multiple model runs
- User definable using Xml
OBJECT USER INTERFACE (OUI)
3-D Animation
<?xml version="1.0" encoding="UTF-8" ?>

<project name="Upper Rio Grande" xmlAdapterClass="oui.mms.MmsProjectXml">
  <paths>
    <path name="mms_work" path="c:/oui_projects/rio_grande/riogr_mms_work" />
    <path name="shapes" path="c:/oui_projects/rio_grande/riogr_oui/shapes" />
    <path name="log" path="c:/oui_projects/rio_grande" />
  </paths>
  <tree name="Upper Rio Grande Project" path="log" logFile="rio_grande.log">
    <node name="Basin Maps"/>
    <node class="oui.treetypes.OuiShapeTreeNode" desc="Subbasin outline" name="Subbasins" />
    <node class="oui.treetypes.OuiShapeTreeNode" desc="Streams" name="Streams" color="blue" theme="streams_v" />
    <node class="oui.treetypes.OuiShapeTreeNode" desc="Cities" name="Cities" color="green" theme="cities" />
    <node class="oui.treetypes.OuiShapeTreeNode" desc="States" name="States" />
    <node class="oui.treetypes.OuiGridTreeNode" desc="El Paso 1000 meter DEM" name="DEM" />
    <node class="oui.treetypes.OuiGridTreeNode" desc="El Paso 1000 meter Slope" name="Slope" />
    <node class="oui.treetypes.OuiGridTreeNode" desc="El Paso 1000 meter Aspect" />
    <node name="Models & Data"/>
  </tree>
</project>
LEVELS OF MODULAR DESIGN

- PROCESS
- MODEL
- FULLY COUPLED MODELS
- LOOSELY COUPLED MODELS
- RESOURCE MANAGEMENT DECISION SUPPORT SYSTEMS
- ANALYSIS AND SUPPORT TOOLS
Simulation 1.1

Data
- Raw
- Processed
  - Resampled DEM
  - Filled DEM
- Climate Data
- LULC
- Veg
- Soil

Model Dimension
- nhru

Model Input

Model Output
- nhru_slope
- hru_slope
- hru_elev
- soil_moist
- pkwater_equi
- transp_beg
- hru_psta

GIS Parameters
- GIS Attributes
- GIS Derivation Equations

GIS Attributes
- Elev
- Slope
- Temp
- Precip

Attributes
- ntemp
- nmonths

Maps

Processing Data
- QC Climate Data
- nttemp
- nmonths

Non-GIS Parameters
- Non-GIS Attributes
- Non-GIS Derivation Equations

Non-GIS Attributes
- dday_slope
- e
- jh_coef

Monthly Streamflow
- Monthly Precip

Attributes
- Maps
- GIS Parameters

Driving Data
- tmax
- tmin
- tsta_elev

Maps

Maps

Dimension
- Model Input

Executables
- PRMS
- ESP
- LUCA

Run Modes
- OMS

Analysis
- DMS
- ESP
- LUCA
GIS WEASEL

- Only requires elevation Grid as input
- Interactively delineate
  - Area of Interest
  - Many kinds of features
    - Streams
    - Elevation bands
    - Landuse
    - Contributing areas
    - Topographic index
    - ......
DIGITAL DATABASES

Vegetation Type (USFS)
Vegetation Density (USFS)
Land Use-Land Cover (USGS)
STATSGO Soils (USDA)
Satellite SW Radiation (UMd)
Monthly PET
AUTOMATED PARAMETER ESTIMATION
GIS WEASEL

• 200+ methods available
• Easily add custom methods
• Configure recipes
• Apply to feature maps
• Exploit many types of data
• Produce maps and ASCII files of parameters
Integrated Modeling and Visualization Tools

- Project Browser
- Simulation Navigator
- Map Window
- Time Slider
- Equation Box
- Data Table
Visualization and Analysis Tool
ANALYSIS and SUPPORT TOOLS
## Multi-step Calibration

<table>
<thead>
<tr>
<th>Step</th>
<th>Calibration data set associated with PRMS state</th>
<th>PRMS parameters sensitive to model state</th>
<th>Parameter Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mean monthly Solar Radiation</td>
<td>dday_intcp</td>
<td>Intercept in temperature degree-day relationship</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dday_slope</td>
<td>Slope in temperature degree-day relationship</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tmax_index</td>
<td>Index temperature used to determine precipitation adjustments to solar radiation</td>
</tr>
<tr>
<td>2</td>
<td>Mean monthly PET</td>
<td>jh_coef</td>
<td>Air temperature coefficient used in Jensen-Haise PET computations</td>
</tr>
<tr>
<td>3</td>
<td>Annual water balance</td>
<td>adjust_rain</td>
<td>Precipitation adjust factor for rain days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>adjust_snow</td>
<td>Precipitation adjust factor for snow days</td>
</tr>
<tr>
<td></td>
<td></td>
<td>psta_nuse</td>
<td>Binary indicator for using station in precipitation distribution calculations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>psta_freq_nuse</td>
<td>Binary indicator for using station in precipitation frequency calculations</td>
</tr>
<tr>
<td>4</td>
<td>Daily flow components: 1. peak flows 2. low flows 3. all daily flows</td>
<td>adjmix_rain</td>
<td>Factor to adjust rain proportion in mixed rain/snow event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tmax_afrain</td>
<td>If HRU maximum temperature exceeds this value, precipitation assumed rain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tmax_alsnow</td>
<td>If HRU maximum temperature is below this value, precipitation assumed snow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tsta_nuse</td>
<td>Binary indicator for using station in temperature distribution calculations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cecn_coef</td>
<td>Convection condensation energy coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>emis_noppt</td>
<td>Emissivity of air on days without precipitation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>freeh2o_cap</td>
<td>Free-water holding capacity of snowpack</td>
</tr>
<tr>
<td></td>
<td></td>
<td>potet_sublim</td>
<td>Proportion of PET that is sublimated from snow surface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smidx_coef</td>
<td>Coefficient in non-linear contributing area algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smidx_exp</td>
<td>Exponent in non-linear contributing area algorithm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gwflow_coef</td>
<td>Groundwater routing coefficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ssrcoef_sq</td>
<td>Coefficient to route subsurface storage to streamflow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>soil2gw_max</td>
<td>Maximum amount of soil water excess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>soil_moist_max</td>
<td>Maximum available water holding capacity of soil profile</td>
</tr>
<tr>
<td></td>
<td></td>
<td>soil_rechr_max</td>
<td>Maximum value for soil recharge zone</td>
</tr>
</tbody>
</table>
Statistical and graphical sensitivity and uncertainty analysis tools
Forecast Methodologies

- Historic data as analog for the future
  - Ensemble Streamflow Prediction (ESP)

-Synthetic time-series
  - Weather Generator

- Atmospheric model output
  - Dynamical Downscaling
  - Statistical Downscaling
Ensemble Streamflow Prediction

Using history as an analog for the future

Simulate to today

Probability of exceedence

Predict future using historic data

NOAA
USGS
BOR
Manager selected 10, 50, and 90 % probability of exceedance
ESP Trace Analysis

- Ensemble Streamflow Prediction (ESP) Tool
- Statistical analysis
- Report generation
Graphical Analysis Tools

- Canned analysis configurations
- Analysis of output data
  - Graphs
    - TimeSeries Plot
    - XY Error
    - Flow duration
    - Residuals
  - Statistics
- Table IO/API
Integrated Adaptive-Modeling and Decision-Support System Summary

- Facilitates multi-disciplinary integration of models and tools to address the issues of water and environmental-resource management.
- Allows rapid evaluation of the effects of decision and management scenarios.
- Allows incorporation of continuing advances in physical, social, and economic sciences.
- Provides an effective means for sharing scientific understanding with stakeholders and decision makers.
- Open source software design allows many to share resources, expertise, knowledge, and costs.
FOCUS ISSUES ARE ADDRESSED THROUGH COLLABORATIVE MULTIDISCIPLINARY EFFORTS
UNIVERSITY COLLABORATION

• University of Colorado
  • NOAA funded -- Coupling of hydrologic and atmospheric models to provide short- to long-term forecasts

• Colorado State University
  • Interagency MOU funded – OMS, Parameter estimation, optimization, and sensitivity analysis tools

• Friedrich Schiller University
  • Collaborative funding – Watershed process modules and modular modeling frameworks

• University of Trento
  • EU funding – Training, Geotop, NewAge, JGrasstools
Water Supply Forecasts

Distributed by the Natural Resources Conservation Service (NRCS)

http://www.wcc.nrcs.usda.gov/wsf

Combined product of NRCS and National Weather Service Forecasts

Prepared by USDA, Natural Resources Conservation Service
National Water and Climate Center
Portland, Oregon
http://www.wcc.nrcs.usda.gov
Working with the Natural Resources Conservation Service (NRCS) to develop a streamflow forecasting toolbox using OMS and PRMS.
The USDA Conservation Effects Assessment Project (CEAP)

A Cooperative Effort to Assess Environmental Effects and Benefits from Conservation Programs at National and Watershed Scales
U.S. MULTI-AGENCY MEMORANDUM OF UNDERSTANDING

Facilitates cooperation in R&D of multi-media environmental models, frameworks, and databases for use in human and environmental health risk assessment

- Nuclear Regulatory Commission (NRC)
- Army Corp of Engineers (COE)
- US Geological Survey (USGS)
- NOAA
- Department of Homeland Security
- Environmental Protection Agency (EPA)
- Department of Energy (DOE)
- US Agricultural Research Service (ARS)
- Natural Resources Conservation Service (NRCS)
Website Information http://sites.google.com/a/environmental-modeling.org/environmental-modeling/Home

- MOU and addendums
- Workshop proceedings
- ISCMEM meeting minutes
- Working group proposals/plans
- Announcements of upcoming meetings
- Contact info
International Collaboration

- Friedrich Schiller University, Jena, Germany
- Chinese Academy of Sciences, Geography and Natural Resources Institute, Beijing
- Chinese Academy of Sciences, CARRERI, Lanzhou
- International Atomic Energy Agency, Vienna
- University of Bodenkultur, Vienna, Austria
- Public Works Research Institute, Japan
- Korean Water Resources Corp., South Korea
- International Water Management Institute, Sri Lanka
- University Trento, Italy
OMS Summary

• Facilitate multi-disciplinary integration of models and tools to address the issues of water and environmental-resource management.

• Allow rapid evaluation of the effects of decision and management scenarios.

• Allow incorporation of continuing advances in physical, social, and economic sciences.

• Provide an effective means for sharing scientific understanding with stakeholders and decision makers.

• Are open source.
“A fool with a tool is still a fool.”
OMS3 Overview
Object Modeling System

• Reduce redundancy in model development
• Improve model code quality
• Simplify model technology transfer
• Allow simulation traceability, support auditing
• Improve model maintainability
• Flexible use and rigorous testing of models
• Allow flexible change in science
• Open source

• Software Framework for Environmental Modeling
  • “Science building blocks”
OMS3 Features

- Lightweight, non-invasive, component-based modeling framework
- Dataflow driven, Multithreaded execution
- Multi-language support
- Conversions / Transformation SPI
- (Ontologies)
- DSLs for Simulations, Calibration, Uncertainty, and Sensitivity analysis
- Auditing, Traceability, Doc generation
- EMF interoperability
### History of the OMS development

<table>
<thead>
<tr>
<th>Year</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001/02</td>
<td>Initial Prototype&lt;br&gt;Swing based</td>
</tr>
<tr>
<td>2003/04</td>
<td>Netbeans Platform&lt;br&gt;Refactoring&lt;br&gt;Central Version Control&lt;br&gt;Standardize Component API</td>
</tr>
<tr>
<td>2005/06</td>
<td>Modeling Projects&lt;br&gt;USDA CoLab Use Project Plan/Process</td>
</tr>
<tr>
<td>2007/09</td>
<td>Calibration Tools&lt;br&gt;Sensitivity Analysis&lt;br&gt;Uncertainty Analysis&lt;br&gt;Data Provisioning&lt;br&gt;Agency Governance</td>
</tr>
<tr>
<td>2010/12/13</td>
<td>OMS3&lt;br&gt;COSU&lt;br&gt;CSIP&lt;br&gt;Cloud Computing&lt;br&gt;Parallelization</td>
</tr>
</tbody>
</table>

- **Netbeans Platform**
- **Refactoring**
- **Central Version Control**
- **Standardize Component API**
- **Modeling Projects**
- **USDA CoLab Use Project Plan/Process**
- **Calibration Tools**
- **Sensitivity Analysis**
- **Uncertainty Analysis**
- **Data Provisioning**
- **Agency Governance**
- **OMS3**
- **COSU**
- **CSIP**
- **Cloud Computing**
- **Parallelization**

**Other Tools and Technologies:***

- **PRMS WSF**
- **CEAP**
- **PRMS - Prototype**
- **J2000(S)**
- **PRMS WSF, WWEM, Range, McCabe MWB**
- **CEAP-AgEs, AgesUI, PRMS WSF, WWEM, Range, McCabe MWB**
- **WASP-Hydrus 1D**
- **Hymod, (GeoTop)**
- **IAEA Rusle2**
OMS3 and other IMFs

Jaegers B. et al., 2010 Proc IEMSS, Ottawa, Canada
OMS Console Versions

1.0

2.2

3.0
## Example Measures McCabe WBM

<table>
<thead>
<tr>
<th>Language/Framework</th>
<th>Total LOC</th>
<th>Average CC/method</th>
<th>Total CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORTRAN only</td>
<td>244</td>
<td>3.33</td>
<td>40</td>
</tr>
<tr>
<td>OMS 3.0 Java</td>
<td>295</td>
<td>2.38</td>
<td>31</td>
</tr>
<tr>
<td>Java only</td>
<td>319</td>
<td>2.85</td>
<td>37</td>
</tr>
<tr>
<td>C++ only</td>
<td>405</td>
<td>2.41</td>
<td>41</td>
</tr>
<tr>
<td>OMS 2.2 Java</td>
<td>450</td>
<td>1.18</td>
<td>103</td>
</tr>
<tr>
<td>ESMF 3.1.1 C</td>
<td>583</td>
<td>1.97</td>
<td>65</td>
</tr>
<tr>
<td>ESMF 3.1.1</td>
<td>683</td>
<td>1.44</td>
<td>56</td>
</tr>
<tr>
<td>FORTRAN</td>
<td>683</td>
<td>1.44</td>
<td>56</td>
</tr>
<tr>
<td>OpenMI 1.4 Java</td>
<td>880</td>
<td>1.61</td>
<td>116</td>
</tr>
<tr>
<td>CCA 0.6.6 Java</td>
<td>1635</td>
<td>2.25</td>
<td>276</td>
</tr>
</tbody>
</table>

[Lloyd, 2011]
Main OMS3 concepts

- Simulations
- Components / Model
- DSL (Groovy/Java)
- Java / FORTRAN / C / C++
Component Class Design

Modeling Component
= POJO + Annotations

Plain Old Java Object

Meta data provide execution control and connectivity, execution support, and documentation/repository support
Simple example “plain old java object” POJO

```java
public class CircleArea {
    public double radius;
    public double area;

    public void runme() {
        area = Math.PI * radius * radius;
    }
}
```
**POJO + Annotations**

*Simple example component with annotated I/O fields.*

```java
public class CircleArea {

    // Tag the fields being used for input and output with @In and @Out
    @In   public double radius;
    @Out  public double area;

    // Fields can have any data type (primitive, custom, or parameterized types) and should be public.
    @Execute
    public void runme() {
        area = Math.PI * radius * radius;
    }
}

///
/// Tag the fields being used for input and output with @In and @Out
/// Fields can have any data type (primitive, custom, or parameterized types) and should be public.
/// No arguments for @In and @Out
/// Required Meta data
/// Applies similar to languages such as FORTRAN, C, C++
```
Building Models

class GroundwaterN {
    @Out
    public HRU hru;
    ...
}

class PlantGrowthStress {
    @In
    public HRU hru;
    ...
}

... out2in(intc, “hru”, pgs);
...

... out2in, field2in, out2field, feedback, ..
# Annotations

<table>
<thead>
<tr>
<th>Component</th>
<th>Field</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>@Description</td>
<td>@Description</td>
<td>@Execute</td>
</tr>
<tr>
<td>@Author</td>
<td></td>
<td>@Initialize</td>
</tr>
<tr>
<td>@Bibliography</td>
<td>@Unit</td>
<td>@Finalize</td>
</tr>
<tr>
<td>@Status</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@VersionInfo</td>
<td>@In</td>
<td></td>
</tr>
<tr>
<td>@SourceInfo</td>
<td>@Out</td>
<td></td>
</tr>
<tr>
<td>@Keywords</td>
<td>@Range</td>
<td></td>
</tr>
<tr>
<td>@Label</td>
<td>@Role</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@Bound</td>
<td></td>
</tr>
<tr>
<td></td>
<td>@Label</td>
<td></td>
</tr>
</tbody>
</table>
Table View (Filter for ‘nhru’ dimensioned parameter)

<table>
<thead>
<tr>
<th>carea_...</th>
<th>cov_type</th>
<th>covden_...</th>
<th>covden_...</th>
<th>dprst_p...</th>
<th>ground...</th>
<th>hru_area</th>
<th>hru_dep...</th>
<th>hru_elev</th>
<th>hru_gwr...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.20000</td>
<td>3</td>
<td>0.10000</td>
<td>0.10000</td>
<td>0.0</td>
<td>0.0</td>
<td>10650.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.20000</td>
<td>3</td>
<td>0.20000</td>
<td>0.20000</td>
<td>0.0</td>
<td>0.0</td>
<td>5540.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.20000</td>
<td>1</td>
<td>0.20000</td>
<td>0.20000</td>
<td>0.0</td>
<td>0.0</td>
<td>13340.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0.20000</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.0</td>
<td>0.0</td>
<td>30070.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0.20000</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.0</td>
<td>0.0</td>
<td>7450.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0.20000</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.0</td>
<td>0.0</td>
<td>5280.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0.20000</td>
<td>3</td>
<td>0.34999</td>
<td>0.34999</td>
<td>0.0</td>
<td>0.0</td>
<td>7940.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0.20000</td>
<td>3</td>
<td>0.40000</td>
<td>0.40000</td>
<td>0.0</td>
<td>0.0</td>
<td>10530.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5490.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>3830.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0.20000</td>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.0</td>
<td>0.0</td>
<td>8610.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0.20000</td>
<td>3</td>
<td>0.34999</td>
<td>0.34999</td>
<td>0.0</td>
<td>0.0</td>
<td>7310.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0.20000</td>
<td>3</td>
<td>0.40000</td>
<td>0.40000</td>
<td>0.0</td>
<td>0.0</td>
<td>14160.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>0.20000</td>
<td>3</td>
<td>0.40000</td>
<td>0.40000</td>
<td>0.0</td>
<td>0.0</td>
<td>12370.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0.20000</td>
<td>3</td>
<td>0.34999</td>
<td>0.34999</td>
<td>0.0</td>
<td>0.0</td>
<td>6820.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0.20000</td>
<td>3</td>
<td>0.34999</td>
<td>0.34999</td>
<td>0.0</td>
<td>0.0</td>
<td>9840.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0.20000</td>
<td>3</td>
<td>0.34999</td>
<td>0.34999</td>
<td>0.0</td>
<td>0.0</td>
<td>14410.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2970.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>5860.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>4170.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2120.0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Statistics on Selection

Auto Documentation

- Docbook5
  - Model meta data
  - Components meta data
  - Parameter sets
  - Simulations

- Docbook5 →
  - PDF
  - HTML
  - DOC(X)
  - …
Main OMS3 concepts

- Simulations
- Components / Model
  - DSL (Groovy/Java)
  - Java / FORTRAN / C / C++
DSLs for simulation

- Lightweight model ‘orchestration’ layer
- Based on language level Builder Pattern
- Descriptive and programmatic
- Scalable for various modeling applications
- Flexible, domain expert can understand it
Simulation DSL
- Descriptive expression of simulations
- Thin layer on top of simulation engines
- Language scripts vs. DSLs, dividing line?

Other Application Examples
- Anti-malaria drug resistance simulations
- Insurances, Financial Institutions, Loan processing
- Nuclear safety simulations
Simulation

Simulation =
Executable Model +
Input Data +
Execution Method +
Context +
Input management
### sim{} elements

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Sub-Element 1</th>
<th>Sub-Element 2</th>
<th>Sub-Element 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim{}</td>
<td>build</td>
<td>components{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>model{}</td>
<td>connect{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>feedback{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>parameter{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>logging{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>resource</td>
<td>outputstrategy{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>output</td>
<td>summary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>efficiency</td>
<td>analysis{}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>timeseries{}</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>calc{}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scatter{}</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>calc{}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flowduration{}</td>
<td>y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>calc{}</td>
</tr>
</tbody>
</table>
Creating and Connecting

Sim s = new Sim();
    s.setName("Drake");
    s.setTiming(true);
    s.setSanitychecks(false);
OutputDescriptor o = new OutputDescriptor();
    o.setDir(work + "/output");
    o.setScheme(SimConst.SIMPLE);
    s.setOutputDescriptor(o);
Model m = new Model();
    m.setClassname("ages.Main");
    s.setModel(m);
Params p = new Params();
    p.setFile(work + "/data/drake.csv");
    m.setParams(p);
    ...

Configuring

Sim s = new Sim();
    s.setName("Drake");
    s.setTiming(true);
    s.setSanitychecks(false);
OutputDescriptor o = new OutputDescriptor();
    s.setOutputDescriptor(o);
    o.setDir(work + "/output");
    o.setScheme(SimConst.SIMPLE);
Model m = new Model();
    m.setClassname("ages.Main");
        s.setModel(m);
Params p = new Params();
    p.setFile(work + "/data/drake.csv");
    m.setParams(p);
....
import oms3.dsl.*;
import oms3.*;

Sim s = new Sim();
   s.setName("Drake");
   s.setTiming(true);
   s.setSanitychecks(false);
OutputDescriptor o = new OutputDescriptor();
   s.setOutputDescriptor(o);
   o.setDir(work + "/output");
   o.setScheme(SimConst.SIMPLE);
Model m = new Model();
   m.setClassname("ages.Main");
   s.setModel(m);
Params p = new Params();
   p.setFile(work + "/data/drake.csv");
   m.setParams(p);
....
Simulation Builder

```plaintext
sim(name:"Drake", timing:true, sanitychecks:false) {
    outputstrategy(dir:"$work/output", scheme:SIMPLE)
    model(classname:"ages.Main") {
        parameter (file:"$work/data/drake.csv")
    }
}
```

- Creation via DSL key words
- Containment handled via scopes
- Configuration via properties
- No “language noise”
Language level Builder Pattern

• Simple mechanism to create a tree-structured graph
• Syntax – level builder pattern (GoF)
• Internally: Chained method invocation, closures, parentheses omission

• Existing Builders
  • Object Graphs, HTML, Swing, Ant, CLI, JMX, …

  -> Simulation Builder
Sim (1)

```java
sim(name:"Efcarson") {
   // workspace directory
def work = System.getProperty("oms3.work");

   // define output strategy: output base dir and
   // the scheme NUMBERED|SIMPLE|DATE
   outputstrategy(dir:"$work/output", scheme:NUMBERED)

   // for class loading: model location
   resource "$work/dist/*.jar"

   // define models
   model(classname:"model.PrmsDdJh") {
      // parameter
      parameter (file:"$work/data/efcarson/params.csv") {
         inputFile  "$work/data/efcarson/data.csv"
         outFile    "out.csv"
         sumFile    "basinsum.csv"
         out        "summary.txt"
         startTime "1980-10-01"
         endTime   "1986-09-30"
      }

   } // logging
      "Strmflow" "INFO"
   }
// ...
```
...  
// model efficiency (optional)  
**efficiency**(obs:"runoff[0]", sim:"basin_cfs", methods:NS+ABSDIF+RMSE, file:"sum1.txt")

// compute some summary for runoff 'on-the-fly' (optional)  
**summary**(time:"date", var:"basin_cfs", moments:MEAN+MIN, period:MOMTHLY, file:"sum.txt")  
**summary**(time:"date", var:"basin_gwflow_cfs", moments:MEAN+MIN+MAX+LAG1, period:YEARLY, file:"sum.txt")

**output**(time:"date", vars:"basin_gwflow_cfs,basin_cfs,runoff[0]", fformat:"7.5f", file:"out1.csv")

**analysis**(title:"Simulation Output"){
    **timeseries**(title:"East Fork Carson", view:COMBINED){  
        **x**(file:"%last/out1.csv", column:"date")  
        **y**(file:"%last/out1.csv", column:"basin_cfs")  
        **calc**(eq:"sim - obs"){
            **sim**(file:"%last/out1.csv", column:"basin_cfs")  
            **obs**(file:"%last/out1.csv", column:"runoff[0]")
        }
        **y**(file:"%last/out1.csv", column:"runoff[0]")
    }
}
Ensemble streamflow prediction: esp

```java
esp(name:"EFCarson") {

    // define output strategy: output base dir and
    // the strategy NUMBERED|SIMPLE|DATE
    outputstrategy(dir: "$work/output", scheme:NUMBERED)

    // for class loading: model locations
    resource "$work/dist/*.jar"

    // define model
    model(classname:"model.PrmsDdJh") {
        //..
    }

    // the number of forecast days after the end of the simulation period
    forecast_days 15
    // forecast_end  "1984-10-15"

    // historical years for to be used for traces
    // years are inclusive
    first_year 1981
    last_year 1983
    analysis(title:"Trace analysis") {
        // relative path name, last output
        esptraces(title:"test 2", dir:"%last", var:"basin_cfs")
    }
}
```
// list of watersheds to analyze
watersheds = ["YampaStmBt", "East", "Animas"]

/////////// Implementation

// base output folder for todays run
today = new File(batch_out, SimConst.now(date_pattern))

// create a simulation builder for all watersheds
sim = new SimBuilder().esp(name:"ALL") {
  analysis(title:"Batch trace analysis: $today") {
    watersheds.each {
      esptraces(title:it, dir:"$today/$it/out", var:"basin_cfs", report:"$today/${it}.txt")
    }
  }
}

sim.graph()
OMS3 Simulations

- Simulations = Model applications with data
- Calibration
  - Shuffled Complex Evolution (SCA) Dynamically Dimensioned Search (DDS), LUCA (MO SCE),
- Sensitivity Analysis
  - Morris Screening, Extended FAST, ..
- Uncertainty Analysis
  - GLUE, Bayesian Monte Carlo, ..
- Forecasting
  - Ensemble Streamflow Prediction
feature rich scripting editor:
- simulation templates
- syntax coloring for OMS3
- command completion on modules
- editor features like word completions, find, replace, IDE-like shortcuts for advanced and faster usage
MORE INFORMATION

• http://oms.javaforge.com/