Jul 1st, 12:00 AM

The Migration of the UTHBAL Hydrologic Model into OpenMI

A. Loukas

K. Kokkinos

L. Vasilides

A. Liakopoulos

Follow this and additional works at: https://scholarsarchive.byu.edu/iemssconference

https://scholarsarchive.byu.edu/iemssconference/2008/all/234

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
The Migration of the UTHBAL Hydrologic Model into OpenMI

A. Loukas\(^a\), K. Kokkinos\(^b\), L. Vasiliades\(^a\), and A. Liakopoulos\(^a\)

\(^a\) Department of Civil Engineering, University of Thessaly, 38334 Volos, Greece (aloukas/lvassil/aliakop@civ.uth.gr)

\(^b\) Department of Information Technology and Telecommunications, TEI, Larissa, Greece (k_kokkinos@teilar.gr)

**Abstract:** This paper deals with the implementation and the migration into OpenMI of a monthly conceptual hydrological model, called UTHBAL. The model has been developed by Loukas et al. [2003, 2007] and now is implemented under the Microsoft Visual Studio .NET C# IDE. The integrated decision support framework provided consists of an autonomous model engine, a graph editor, an optimization/calibration server and a migration wrapper. The software allows input data polymorphism and extensibility of model integration, which is lumped in the present form. The infrastructure for incorporating various optimization libraries has also been built. Online data exchange is feasible since the model has been migrated into OpenMI through a dynamically configurable wrapper. We found that the model characteristics relevant to coupling with other hydrological models lie on the space, time, structure and optimization compatibilities. Thus other models which analyze similar phenomena can seamlessly be coupled with ours.

**Keywords:** Hydrological Modelling; Migration; Rainfall-Runoff Models; OpenMI.

1. INTRODUCTION

Hydrological modelling and decision support frameworks are intended to be used by water resource managers and modelers in water resources management. Unfortunately, the existing frameworks only attempt to combine their sub-module codes into a single program, requiring the translation of different source codes with different compiler versions. Such a process has been standard practice within the hydrologic community, however, it is costly and time consuming. The idea of external interaction of hydrological software has recently been established through the development of the Open Modeling Interface (OpenMI) standard [Gregersen et al., 2007; Moore and Tindall, 2005]. The use of the OpenMI-standard not only address the coupling of individual catchment processes but also guarantees minimal programming effort for the spatial and temporal synchronization and data interaction between compatible models. The above methodology, however, can be applied only by implementing the necessary infrastructure in order to achieve compliance with OpenMI. This means that every model must be accompanied by a migration wrapper in order to interact with external models through the OpenMI standard [Gregersen et al., 2005].

This study describes the migration process of the monthly conceptual hydrological model UTHBAL for water balance modeling [Loukas et al., 2003; Loukas et al., 2007]. Lumped water balance models have been developed at various time scales (e.g. hourly, daily, monthly and yearly) and to varying degrees of complexity. Monthly water balance models were first developed in the 1940s by Thornthwaite [1948] and have been adopted, modified, and applied to a wide spectrum of hydrological problems. Recently, these hydrological models have been employed to explore the impact of climatic change. They
also have been utilized for long-range streamflow forecasting. Although such applications may use hourly or daily models, these models are, however, more data intensive and have more parameters than the monthly models. A complete review of water balance model applications could be found in Xu and Singh [1998].

Before the migration process, the model was implemented under the Microsoft Visual Studio .NET C# IDE. The framework consists of four basic components: (a) the data incorporation component, (b) the optimization-verification component, (c) the graphical component and (d) the migration component. The above modules have been implemented to be portable and externally executable as independent dynamic link libraries (dlls). For the creation of the appropriate OpenMI wrapper engine, a semiautomatic approach is used: the user is asked to identify the preferable hydrologic components to be exchanged in the interaction process by formulating various .omi files. The migration process is semi-dynamic, thus helping the modeler to provide various OpenMI migration wrappers according to the coupling characteristics and dimensionalities with other models. This migration process has been verified under the NUnit test framework for all created .omi files that can be produced with all possible combinations of interchanging data sets.

The migration process correctness and the coupling ability of our application has been verified for a specific case study involving the coupling of the UTHBAL framework with a groundwater model for simulating surface runoff and groundwater recharge. In the following sections we provide: (a) a description of the UTHBAL model, (b) a software engineering point-of-view explanation of the developed framework summarizing its functional requirements, (c) a rigorous algorithm for the migration wrapper development according to OpenMI standards and (d) a thorough proof of concept of the data interchange process for the above referred case study.

2. THE UTHBAL MODEL

The model has been developed by Loukas et al. [2003] and updated in its present form for simulation of hydrologic cycle components [Loukas et al., 2007]. The UTHBAL model has been successfully applied to watersheds in Cyprus, in the regions of Crete and Thessaly of Greece and in the Greek-Bulgarian transboundary Nestos/Mesta River basin.

![Figure 1. UTHBAL flowchart and model parameters.](image)

The UTHBAL model requires monthly values of mean temperature, precipitation, and potential evapotranspiration and produces values for actual evapotranspiration, soil moisture, groundwater and surface runoff. The model separates the total precipitation into rainfall and snowfall, because the correct division of precipitation is essential for modelling...
the mass balance of seasonal snow covers and for accurate runoff simulation [WMO, 1986]. The rain-snow percentage is estimated using a logistic relationship based on mean monthly temperature [Knight et al., 2001]:

\[
\%S = \begin{cases} 
0 & \text{if } T \geq 12.22 \degree C \\
\frac{1}{(1.35^T \cdot 1.61) + 1} & \text{if } -10^\circ C \leq T \leq 12.22 \degree C \\
1 & \text{if } T \leq -10^\circ C
\end{cases}
\]

where, \(\%S\) is the monthly percentage of precipitation which is falling in the form of snow and \(T\) the mean monthly temperature. The snowmelt of month \(J\), \(SM(J)\) is estimated using the simple degree-day method [Semadeni-Davies, 1997]:

\[
SM(J) = C_m \cdot T(J)
\]

where, \(T\) is the mean monthly temperature and \(C_m\) is the monthly melt rate factor (mm/\degree C per month).

The snow water equivalent of the accumulated snowpack, \(SWE_{sp}\) is estimated by:

\[
SWE_{sp}(J) = SWE_{sp}(J-1) + S(J) - SM(J)
\]

where, \(S(J)\) is the snow fallen during month \(J\) equals to:

\[
S(J) = \%S \cdot P(J)
\]

where, \(P(J)\) is the total precipitation of month \(J\).

The model divides the total watershed runoff into three components: the surface runoff, the interflow, and the baseflow using a soil moisture mechanism (Figure 1). The first priority of the model is to fulfill the actual evapotranspiration. The monthly actual evapotranspiration depends on the available soil moisture and the average surface potential evapotranspiration for that month. The model estimates and gives as output, apart from the actual evapotranspiration, the available soil moisture, the deep recharge to groundwater, and the surface runoff. The detailed algorithm of UTHBAL could be found in a recent paper [Loukas et al., 2007]

3. THE UTHBAL IN THE OPENMI FRAMEWORK

The UTHBAL model was implemented under the Microsoft Visual Studio .NET C# IDE (Figure 2), in an integrated environment that accepts time series data based on different geographical locations. Currently the manipulation is implemented in lumped mode. The extension into the distributed mode is an ongoing process. This will incorporate interaction and data interchange with GIS applications running these applications inside the UTHBAL framework. The framework is constructed in four basic modules each of those integrating respectively: (a) the XML-format storage of the applications, (b) the calibration and optimization of the model, (c) the visualization of the output data and (d) the automated migration and export into OpenMI. The following subsections elaborate on the above components. An architectural schematic of UTHBAL within the OpenMI framework is depicted in Figure 3.

3.1 Storage Component

Case studies based on the UTHBAL can be stored, loaded, and run separately by the framework. Each case study is stored in a .ubm file, an XML-type format file that includes: (1) a standard geographic location for the lumped mode or the XYZ-polygon coordinates
for the distributed, (2) temporal discretization of the case study (day, month, etc.) and (3) relative paths to the appropriate dlls. The IEngine object is then programmed to read from that XML-file and is prepared to create the set of input/output data for exchange. We also wrap in a .dll the object that does all the intermediate calculations for the simulation of the time series. That makes the running of any UTHBAL - OpenMI simulations fully portable and independent of the physical presence of the UTHBAL application. In the future, we plan to create a web service so users can run their case scenarios using the UTHBAL remotely.

![A class diagram of the UTHBAL model.](image)

### 3.2 Calibration, Optimization and Verification

The UTHBAL model has six parameters to be optimized in order to estimate watershed runoff and hydrological cycle components, namely, Cm, CN, K, α, β, and γ. The framework allows the user to calibrate the model by dynamically selecting any preferable time period for calibration. Currently only the Generalized Reduced Gradient method [Lasdon and Smith, 1992] is used for optimizing model parameters and Model Efficiency ($Eff$) [Nash and Sutcliffe, 1970] was used as the objective function between the observed and simulated runoff. Apart of the $Eff$, two more statistical measures of the quality of runoff simulation are used: a) the percentage runoff volume difference ($DV\%$), and b) the coefficient of determination ($R^2$) between the observed and simulated runoff. The inclusion of several other optimization methods to the framework is an ongoing process also. Our intention is to extend the current module into an optimization server, thus making the framework suitable for multi-threaded applications, serving multiple users concurrently.

### 3.3 Output Data Visualization

The UTHBAL framework applies a unique visualization method of input and output data implemented using C# Crystal Reports and the ZedGraph library [ZedGraph, 2008]. Data
can be visualized in histograms by having the user dynamically select which quantities to depict (Figure 4). The online manipulation such as zooming and the storage of the graphs in various formats (jpg, pdf etc.) is also available (Figure 5). The stored graphs are tightly coupled to the individual case studies by inserting their storage path into the .ubm XML-structure for portability.

Figure 3. The architecture of UTHBAL.

3.4 Model Migration to OpenMI

The provision of data exchange of the UTHBAL framework with other hydrologic models is based on the OpenMI paradigm. However, special effort has been made to extend this
practice by establishing a metadata connection of the model to the OpenMI editor through the use of .ubm files. An .ubm file is simply an XML file carrying out the model inputs and outputs along with the time series specifications. The location of this file for each simulation run is passed to the IEngine component of OpenMI as a parameter to the OMI file. Since any OMI file would need access to the UTHBAL classes for the nodes available, this effectively means that, for our case these class libraries are also ported. The result is that, any user is able to export a UTHBAL case study to an OMI model in one workstation and run the produced OpenMI simulation in another. The migration process accomplishes that by physically copying the necessary libraries from the UTHBAL directory to the target OpenMI directory.

Another user friendly feature of the framework is the automatic built up of the OMI file for each case study. This is accomplished by scripting the standard XML tag information including the locations of the IEngine, the LinkableEngine and the LinkableComponent of the application.

We added to the functional requirements of the framework the ability of the user to select which model variables should be available through the OpenMI data exchange. Since UTHBAL model can be migrated to OpenMI in order to exchange data with other models, it is practical to always show all intermediate output data variables. This is solved by saving the variable names in a binary file in the same directory as the OMI file. The file needs to be present so that the UthBalModelOMIEngine knows the variables to show the user. The LinkableEngine (called UthBalModelOMIInterface) instantiates a new UthBalModelOMIEngine which is simply an implementation of the OpenMI’s IEngine interface. The UthBalModelOMIEngine class looks for a path of the .ubm file so that it can instantiate a new UTHBAL model. If none is found, the UTHBAL model class automatically looks for .ubm files in its own directory. We cannot have a UthBalModelOMIEngine object without the corresponding .ubm file, since the simulation cannot run without data. The UthBalModelOMIEngine class handles all communication from the UTHBAL model to the OpenMI through the procedures of the IEngine interface. More specifically, the time horizon of the model is defined as an ITimeSpan variable (OpenMI’s time period representation) which spans from the first day of the month of the first node to the first day of the month of the last node of the model. The outputExchangeItems array is filled according to the previously mentioned binary file, while the model description (given by using the GetModelDescription method) is set to show the model’s name. Also the GetValues method is set to use a method in the UTHBAL framework that reads a variable from a given node according to the variable name. This method was placed in the node class specifically for facilitating the migration.

Figure 4. Dynamic selection for depicting intermediate output data.  
Figure 5. Resulting graphs.
4. A TESTING CASE OF UTHBAL MIGRATION

The migration process correctness and the coupling ability of our application has been verified for a specific case study involving the coupling of the UTHBAL framework with the groundwater regional model produced by the OpenMI Association. Changes in the argument section of the original RegionalGWModel.omi file included only the setting of simulation starting and ending days. We considered the general case scenario where UTHBAL possibly interchanges all calculated variables with compatible models. Quantity passing succeeded for the entire temporal spread as expected. The promotion of UTHBAL to the distributed phase relatively to its OpenMI linkage is also pre-tested with the inclusion of an array of ILinkableComponents passing only the single value of deep infiltration or groundwater recharge. Figure 6 illustrates the NUnit testing and Figure 7 the coupling testing process.

Figure 6. NUnit testing of UTHBAL wrapper.

Figure 7. Simulation run of the coupling test.

5. CONCLUSIONS

In this paper, the migration process of a monthly conceptual hydrological model in the OpenMI framework has been presented. The hydrological model is the UTHBAL model
developed in the University of Thessaly. Before the migration process, the model was implemented under the Microsoft Visual Studio .NET C# IDE. The framework consists of four basic components: (a) the data incorporation component, (b) the graphical component, (c) the optimization-verification component and (d) the migration component. More specifically, (a) the data incorporation component involves the input of data in text or Microsoft Excel© format, as well as the storage process of case studies in separate XML format type files, (b) the graphical component depicts the calculated time series graphs of the components of hydrologic cycle (i.e. rainfall, snowfall, snowmelt, actual evapotranspiration, soil moisture, surface and groundwater runoff), (c) the optimization component involves the techniques of the model parameter optimization based on Generalized Reduced Gradient method and finally (d) the migration component that implements the model engine wrapping. The above modules have been implemented to be portable and externally executable as independent dynamic link libraries (dlls). For the creation of the appropriate OpenMI wrapper engine, a semiautomatic approach is used: the user is asked to identify the preferable hydrologic components to be exchanged in the interaction process by formulating various .omi files. The migration process has been verified under the NUnit Testing framework for all created .omi files for a specific case study and for the communication of UTHBAL with a groundwater model.

ACKNOWLEDGEMENTS

This research is partially funded by EC under the LIFE Programme and contributing to the implementation of the thematic component LIFE-Environment under the policy area "Sustainable management of ground water and surface water management". Contract no: LIFE06 ENV/UK/000409.

REFERENCES


