8-1-2014

Developing An Integrated Modelling System For Blue-Green Solutions

Christos Makropoulos

Evangelos Rozos

Cedo Maksimovic

11th International Conference on Hydroinformatics
HIC 2014, New York City, USA

DEVELOPING AN INTEGRATED MODELLING SYSTEM FOR BLUE-GREEN SOLUTIONS

CHRISTOS MAKROPOULOS (1), EVANGELOS ROZOS (1), AND CEDO MAKSIMOVIC (2)
(1): School of Civil Engineering, National Technical University of Athens, Heroon Polytechneiou 5, GR-157 80, Athens, Greece
(2): Department of Civil and Environmental Engineering, Imperial College London, Imperial College Road, SW7 2AZ, London, UK

Blue-green interventions represent the next level of integration for sustainable cities: that of an integrated urban water and urban green design, operation and management. The key concept is that a more holistic infrastructure design approach would present a win-win scenario, in which urban green would be utilized as infrastructure for water services (e.g. mitigating urban floods) while urban water infrastructure would be used as a source of irrigation for urban green, increasing their performance in a range of services including amenities, reducing heat island effects and increasing ecosystem services. However, this focus on integration brings into sharp relief another need: that of developing models and tools able to investigate the interactions between different green and blue system elements and processes. This "ecosystem" of models and tools presents a challenge due to its scope, in terms of development, but also the challenge of model integration. This paper discusses these challenges and proposes a three level approach to building an integrated modelling system for this case, which is able to: (a) support in the choice of appropriate models; (b) facilitate their linking in runtime and (c) enable the homogenization of results from the different models into common views supporting decision making. The use of standards, in this case OpenMI, are discussed in the light of the proposed approach. The concept is illustrated using a limited set of simple models developed for blue-green solutions design and the preliminary results are presented and discussed.

INTRODUCTION

It has been argued that the combined effects of climatic changes and increasing urbanization require a paradigm change in the design and operation of urban water systems and urban vegetated areas. The Blue-Green Dream (BGD) approach [1] advocates combining the management of water and green spaces in urban environments to better complement each other, reduce the need for more expensive infrastructure, while improving local environmental conditions (http://bgd.org.uk). Specifically, BGD is about interactions between different physical domains, between biological and engineering systems, between technologies and social systems, etc. One approach to study these complex interactions could be to develop new "monolithic" hyper-models able to simulate "every" aspect of this conceptualization. However, this choice, apart from being a resource-intensive task (challenging in both development and

validation), would also lack flexibility (the practitioner would need to employ a complex model even for simple applications) and adaptability (it would be very difficult to add new features as needs or research progressed), and would have to cope with the diversity of approaches adopted by various researchers involved in the development phase.

A more elegant approach, which could turn this diversity into an advantage, would be to endorse a process that would focus on and develop tools at each component level, but then ensure link-up when running the integrated model – and indeed enable different levels of complexity of integrated models to be available for ad hoc combination by the problem owner. To achieve this, one needs to ensure interoperability, seamless communication and exchange of data (possibly in real/run time) between different tools developed by different people. This can be accomplished by adopting *common standard interfaces*, which will allow the developers to progress with their work "under the hood" independently. A side benefit of this approach is that each tool developed using these common standards would be able to talk not only to other BGD tools/models, but also to everything else that complies to these standards (e.g. software by USEPA, ESRI, DHI, DELTARES, MWHSoft, etc). According to Brandmeyer and Karimi [2], the use of standard interfaces between models belongs to the highest of the five levels of classification they use to characterize the degree of integration.

Exactly such a common standard interface is OpenMI (Open Modeling Interface) [3], which has been developed with the purpose of being the glue that can link together model components from various origins. OpenMI provides a standardized interface to define, describe and transfer data on a time basis between software components that run simultaneously.

This modularity, enabled by the OpenMI framework, allows users to easily build compositions of models to study for example, the multifaceted aspects of a BGD approach to infrastructure design. This modularity will of course also significantly increase the number of modelling options, for prospective model users and BGD practitioners and with this increase come questions such as, for example, "which are the most suitable models a specific problem at hand" and "how can I aggregate and present results from different models so that the impact of a potential solution is intuitively presented to decision makers"?

This paper offers some preliminary suggestions addressing these two questions along with brief guidelines on how to implement the OpenMI standard. The paper follows the modelling process, starting from making models linkable, to selecting models to link, to running a composition of linked models, to visualising results based on examples from the BGD toolbox.

ENABLING MODEL LINKING

OpenMI allows the linking of models with different spatial and temporal representations: for example, it supports the linking of river models to groundwater models, where the river model typically uses a one-dimensional grid and a short timestep and the groundwater model uses a two- or three-dimensional grid and a longer timestep [4]. The OpenMI 2.0 Standard (latest

release) was followed by the release of FluidEarth[1]. FluidEarth (current version 2.2) is an open source implementation of the OpenMI 2.0 interface standard for data exchange between numerical models during run-time. It consists of the FluidEarth Software Development Kit (SDK) to assist with making models and components OpenMI 2.0 compliant, and the Pipistrelle GUI for linking and running OpenMI compliant models and components using one-way or two-way links for the exchange of data [5].

OpenMI is designed to accommodate the easy migration of existing modelling systems since their re-implementation may not be economically feasible due to the large investments that have been put into their development and testing. For legacy code, wrapping will most often be the technological choice to migrate to OpenMI. This means that an existing model engine (i.e. a computational core, developed in Fortran or C or any other language that produces native code) is encapsulated in a so-called wrapper that meets the interface specification of an OpenMI linkable component [6]. The steps that to be followed to wrap an existing model are:

1. Prepare the DLL that contains the core engine of the model.
2. Prepare the OpenMI wrapper, i.e. the C# classes that wrap the engine model and implement the keys required by the OpenMI standard.
3. Prepare a tool to facilitate the compilation of model projects into OMI files.

OMI files are XML files that provide information required to run an OpenMI component (parameters, the initial values, and the topology). Normally, for stand-alone applications of each model, this information is contained in the files that form what is often termed a "project" of the model. However, Pipistrelle (which is the tool used to run an OpenMI composition) is, understandably, not able to read the native project format of every model involved. For this reason, OMI files are used to store information in a standardized manner. This means that whereas the OpenMI wrapper for a given model is prepared by the developer, OMI files are prepared by the user for each specific case study. For this reason a tool that facilitates the preparation of OMI files (i.e. the conversion of model projects into OMI files) should be provided by the developer. More detailed information on the implementation of the OpenMI standard for BGD models can be found at [7].

SELECTING APPROPRIATE MODELS TO LINK

The flexibility provided by OpenMI allows the creation of a repository of models, which could be used as the basic components to build (integrated modelling) compositions – to develop and evaluate in our case BGD designs. The identification of the most appropriate models for each specific composition needs then be facilitated by what will be termed hereafter a "choice support" system. The basic idea is to enable the user to state the problem using keyword-terms describing the thematic tags which categorize/characterize the case study. Then, the system presents a list of models (within and outside BGD) ranked according to their estimated

[1] http://fluidearth.net/default.aspx. Fluidearth is an HR Wallingford initiative.

suitability for addressing the problem at hand, while also pointing at additional issues, datasets, models and interactions to consider.

The conceptualization of this choice-support system is schematically displayed in Figure 1. The system comprises three basic elements: the knowledge base, the inference engine and the user interface.

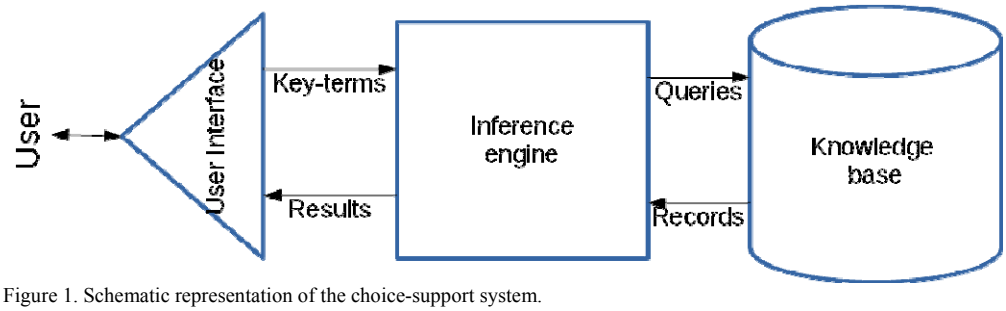

Figure 1. Schematic representation of the choice-support system.

The knowledge base is where all information (specifications, categorization, assumptions, limitations, etc.) about the models contained in the repository are stored. This is could be an information database that provides a much simpler data storage/retrieval mechanism compared to the classic tabular relations used in RDBMS (see NoSQL database [8]). This suggestion has the advantage that the often "cryptic" tables resulting in traditional SQL databases are avoided, all data are kept in the same collection (and hence are directly readable) and the structure of the database can be easily modified at any time with the addition of new models and knowledge.

The inference engine includes the algorithm that translates the terms searched by the user into appropriate queries that are sent to the knowledge base. The inference engine also includes the algorithm that synthesizes the answer of the database into human understandable information.

The last element of the choice-support system is the user interface. This provides the user with controls to type-in, for example, keyword terms and to narrow down the search into relevant areas of interest. The interface also displays the results of the query returned by the inference engine and provides the option to download the models that the user selected as most relevant from the repository – or from remote locations associated with the repository (e.g through hyperlinks).

RUNNING A COMPOSITION OF LINKED MODELS

After the selection of appropriate models, the user needs to build the composition, i.e., define the schema that determines how the models interact with each other. This section gives a simple example of such a composition (Figure 2) with just two UWOT ([9] and [9]) OpenMI components. Of course, in real applications more complicated compositions, in both number and variety of included components, would be employed.

The left component of Figure 2 (named “UWOT Green HHs”) is a UWOT component that simulates the water cycle of a group of identical households that employ Green technologies (see Figure 3), whereas the other component (named “UWOT SUDS”) simulates a central SUDS (see Figure 4). The “UWOT SUDS” receives the runoff output of the “UWOT Green HHs” component. The output of the “UWOT SUDS” is connected to a "Get Values Call" component, which indicates to the composition solver (in this case Pipistrelle) the simulation starting point: the components are run starting from the component directly connected to the "Get Values Call" and proceeding upstream.

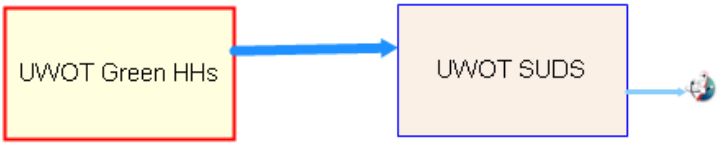

Figure 2: Example of an OpenMI composition that includes two UWOT components.

The inner topology of the two UWOT components (Figure 3 and 4**)** was prepared using UWOT's incorporated CAD interface. The wrapper is responsible for transferring data from the composition connections (see Figure 2) to the input and output nodes of the components' inner topology. The IN element in Figure 4 is fed with data arriving at the input of the “UWOT SUDS” component of Figure 2. The two loggers in Figure 3 (depicted with a cassette-tape symbol) provide data to the output of the “UWOT Green HHs” component of Figure 2.

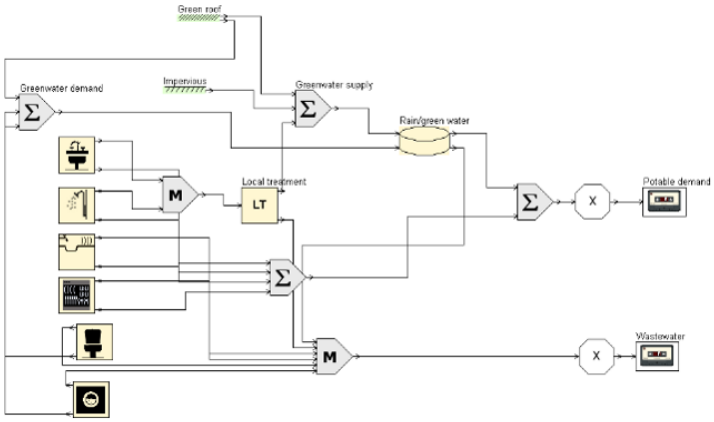

Figure 3: UWOT representation of a group of identical households with BG technologies

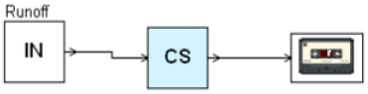

Figure 4: UWOT representation of central SUDS.

It should be noted that both UWOT components in this composition use the same wrapper and engine (these are two DLL files), but have different OMI files. The steps that were followed to prepare this composition were:

1. Prepare the two corresponding UWOT projects (displayed in Figure 3 and Figure 4) using UWOT itself.
2. Export the UWOT projects using the "Export files" feature (File → Export → Export files) to a different folder for each project. This action exports the information of a UWOT project as a series of text files.
3. Use a tool for converting UWOT exported projects to OMI files (the two projects are compiled into two OMI files, HHs_Green.omi and SUDS.omi). Depending on the specifications of this tool, i.e. whether or not it can directly read the native format of a model projects, the previous step can be skipped (this is up to the developer to decide).
4. Place the two OMI files into the same folder with the DLLs of the wrapper and engine.
5. From within Pipistrelle, Right click → Add → Component → navigate to the folder where OMI files and DLLs are and click Open.

The Pipistrelle GUI (installed with FluidEarth) provides a GIS viewer to visualize the spatial relationships between the components. Figure 5 shows the Pipistrelle GIS displaying spatially the composition of Figure 2.

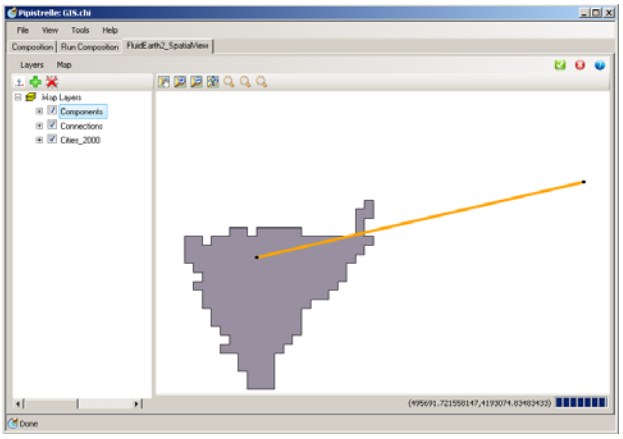

Figure 5: Spatial representation of the connection between the components of Figure 2 composition.

VISUALIZING RESULTS

Visualization of the results of individual models is generally an easy task. However, the visualization of results produced by a series of coupled models, like those employed in a composition that seeks an optimum BGD solution, is more challenging since it must:

1. coherently and comprehensively present results coming from a variety of models.
2. standardize a range of possible (sustainability) criteria to support decisions.
3. perform multi-criteria analysis based on, for example, a predefined set of Performance Indicators (PI).

We suggest that the first step to fulfil the requirements above is to adopt a standardized method of exporting simulation results. Following the concept of OMI files, the suggested method is to use an XML format, whose main element will be termed "indicator" and its attributes will include the following:

- The attribute "name". This is the name of the PI.
- The attribute "value". This is the value of a specific PI corresponding to an assessment of a BGD solution.
- The attribute "reference value". This is the value of a specific PI corresponding to the base scenario.
- The attribute "increase desirable". This indicates whether or not increased values of the indicator are desirable (e.g. an increase in runoff peaks is not desirable whereas an increase of water quality is desirable). This allows downstream models to know how to interpret the information and is important for optimisation purposes.
- The attribute "aggregation method". This provides the aggregation method that will be used in the case where more than one models of the same composition include the same indicator in their evaluation.

With the adoption of this XML file, the wrapper of each component will be able to summarize the results from the model run and export this information into the file. A visualization tool could then plot the information contained in the XML files, by processing every XML file produced from the run of a composition, perform appropriate aggregations wherever this is required and derive a single comprehensive plot for the whole composition. An example of such a plot (in the form of a spider diagram) can be seen in Figure 6.

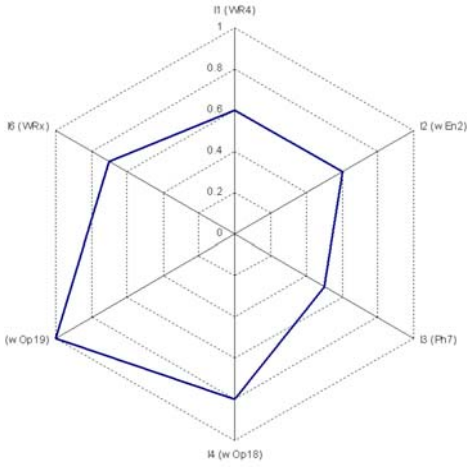

Figure 6: Example of visualization of performance indicators values estimated after a composition run.

CONCLUSIONS

The BGD vision advocates the combined management of water and green spaces in urban environments. This presupposes combined tools and as such a framework to combine dedicated models for studying the various interactions involved in the BGD approach is required. In this paper we propose OpenMI as the appropriate framework and outline the process by which a model can be made OpenMI compliant, providing a simple example from the BGD context. We also discuss issues that arise from the availability of combination of models, i.e., the need for identification of the most suitable models linked to the characteristics of each case study and the need for a comprehensive and coherent visualization of results. For the former, a knowledge management system is suggested as a "choice-support" tool while for the latter, we propose the use of a standardized XML file able to capture simulation results along with a tool developed to plot this file, as part of the decision making process.

Acknowledgments

This work was supported by the Blue Green Dream (BGD) Climate-KIC/FIT Project (bgd.org.uk).

REFERENCES

[1] Maksimovic, C., "Next generation paradigm for urban pluvial flood modelling, prediction, management and vulnerability reduction–Interaction between RainGain and Blue Green Dream projects". *Proc. EGU General Assembly 2012 22–27 April*, Vienna, Austria, (2012), p. 14347.

[2] Brandmeyer, E. and Karimi, H.A., "Coupling methodologies for environmental models", *Environmental Modelling & Software,* Vol. 15, (2000), pp 479-488.

[3] Moore, R.V. and Tindall, C.I., "An overview of the Open Modelling Interface and environment (the OpenMI).", *Environ. Sci. Policy*, Vol. 8, No. 3, (2005), pp 279-286.

[4] Gregersen, J.P., Gijsbers, P.J.A. and Westen S.J.P., "OpenMI: Open modelling Interface.", *Journal of Hydroinformatics*, Vol. 9, No. 3, (2007), pp 175-191.

[5] Sutherland, J., Bolster, M. and Haper, A., "Beachplan as an Open-MI composition," *Proceedings of 2013 IAHR World Congress*, (2013).

[6] Moore et al., "Scope For the OpenMI (Version 2.0)", *The OpenMI Document Series*, (2010).

[7] Rozos, E., S. Kozanis, and C. Makropoulos, "*Integrated Modelling System*", BGD internal project report, (2014).

[8] Wikipedia, (2014), "NoSQL", accessed on March 2014, available online at http://en.wikipedia.org/wiki/NoSQL.

[9] Rozos, E., Makropoulos, C., and Maksimovic, C., (2013), "Rethinking urban areas: an example of an integrated blue-green approach", *Water Science and Technology: Water Supply*, 2013.

[10] Rozos, E., and Makropoulos, C., (2013), "Source to tap urban water cycle modelling", Environmental Modelling and Software, Vol 41, pp 139–150.