Environmental Modelling & Software 73 (2015) 260–271

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

# Using a Model MAP to prepare hydro-meteorological models for generic use

Quillon Harpham

*HR Wallingford, Howbery Park, Wallingford, Oxfordshire, OX10 8BA, United Kingdom*

ARTICLE INFO

*Article history:*
Received 20 May 2015
Received in revised form
5 August 2015
Accepted 8 August 2015
Available online 5 September 2015

*Keywords:*
Environmental modeling
Hydro-meteorology
Model integration
Metadata
Standards

ABSTRACT

Structured environments for executing environmental numerical models are becoming increasingly common, typically including functions for discovering models, running and integrating them. As these environments proliferate and mature, a set of topics is emerging as common ground between them. This paper abstracts common characteristics from leading integrated modelling technologies and derives a generic framework, characterised as a Model MAP – Metadata (including documentation and licence), Adaptors (to common standards) and Portability (of model components). The idea is to form a gateway concept consisting of a checklist of elements which must be in place before a numerical model is offered for interoperability in a structured environment and at a level of abstraction suitable to support environmental model interoperability in general. Following comparison to the Component-Based Water Resource Model Ontology, the Model MAP is applied to DRIHM, an hydro-meteorological research infrastructure, as the initial use case and more generic aspects are also discussed.



## 1. Introduction

Structured environments for executing environmental numerical models are becoming increasingly common. The objectives of these environments are usually to allow models to be more widely available to user communities, to reduce the effort required to prepare the models for use and to provide appropriate computing environments which allow scientists to focus on the science instead of spending the majority of their time battling ICT issues. Such environments are typically built upon computing resources capable of executing a model run in a reasonable timescale and usually incorporate functionality enabling users to discover models and evaluate their suitability, run the models, and chain them together as an integrated system (such as a set of models capable of passing data between them so that they might influence one another). Sometimes facilities are provided to set up the model – by setting arguments and selecting supporting datasets – otherwise the user must prepare their model offline for subsequent upload.

Sutherland et al. (2015) observe that the discipline of integrated environmental modelling is at the stage where systemic knowledge management can be applied to make gains through the application of consolidated standards and approaches as would usually be found in such structured environments. As these environments proliferate and mature, a set of topics is emerging as common ground between them. A key aspect given is the provision of standardised metadata and other supporting information such as guides and manuals describing components required for re-use, both for discovery and use purposes (observed by Michener (2006) with respect to ecological data management). This includes adequate licencing conditions allowing components which have been licenced separately to be handled in a single framework. In managing uncertainty in integrated environmental modelling, Bastin et al. (2013) draw out the aspect of model interface technologies and the frameworks which implement them. Structured methods and standards are used to interface between distinct modelling components as uncertainty is propagated between them.

One such structured environment is the Distributed Research

*Acronyms and Abbreviations:* CF, Climate and Forecasting; CUAHSI-HIS, Consortium of Universities for the Advancement of Hydrological Science Incorporated – Hydrologic Information System; CSDMS, Community Surface Dynamics Modelling System; DRIHM, Distributed Research Infrastructure for Hydro-Meteorology; DRIHM2US, Distributed Research Infrastructure for Hydro-Meteorology to the United States of America; HPC, High Performance Computing; ICT, Information and Communication Technology; NetCDF, Network Common Data Form; OGC, Open Geospatial Consortium; OpenMI, Open Modelling Interface; RIBS, Real-time Interactive Basin Simulator; SDK, Software Development Kit; WaterML, Water Markup Language; WRF-ARW, Weather Research and Forecasting – Advanced Research; XML, Extensible Markup Language.

*E-mail address:* q.harpham@hrwallingford.co.uk.

http://dx.doi.org/10.1016/j.envsoft.2015.08.007
1364-8152/

Infrastructure for Hydro-Meteorology (DRIHM; accessible at http://ww.drihm.eu, https://portal.drihm.eu/ (Grid certificate required for many functions)): an eInfrastructure allowing researchers to formulate and execute hydro-meteorological model chains to study flooding events (D'Agostino et al., 2014 and Danovaro et al., 2014), incorporating the provision of both driving data and numerical models. It is not tied to a single back-end ICT infrastructure and incorporates all of HPC, Grid and Cloud resources through a single portal based around the gUSE workflow engine (Balasko and Farkas, 2011). Each numerical model is given access to the appropriate resources for its execution – for example meteorological models typically utilizing HPC with output data passed down the model chain to hydrological models utilizing Grid resources and hydraulic models typically utilizing the cloud. Also incorporating CUAHSI-HIS – by utilising its web interface to serve heterogeneous point series data – the primary use case of flash flooding extends from meteorology into hydrology, hydraulics and impact (in terms of financial damage and personal injury). These differing model domains require a more generic approach to offering numerical models for formal interoperability. Moreover, all of the models featured in the infrastructure are legacy applications. They range from established numerical models well adopted in their domains to research applications with frequently updated code-bases written by scientific programmers. This variation offers heterogeneity that is, perhaps, uncommon in research infrastructures. Nativi et al. (2013) outline a vision including a set of facilitating principles emphasising access and ease of entry and warn that legacy applications may require considerable modifications in order to be compatible. A similar observation is made by Athanasiadis et al. (2009), who indicate that interoperability issues can play a major role in model integration when the models are developed in different programming languages, platforms and operating systems, as is the case here.

In order to collect these models together and offer them in a common framework it is necessary to provide a highly generic, base level for this provision which is technically agnostic, but then leads towards the more specific standardisation and structure which must be demanded by the lower level technical services and then towards the formal standardisation of the model components. As interoperability between infrastructures for running models becomes more common-place, so the need for a high level, gateway concept which is applicable to many such infrastructures is brought into focus. This concept needs to be accessible to scientific programmers and researchers providing initial steps to model interoperability and standardisation, whilst being lightweight and simple to apply.

Accordingly, the objectives of this paper are to derive this concept as an abstraction of many of the commonalities observed, describe the various aspects and give it a simple characterisation. The idea is to form a checklist of elements which must be in place before a numerical model is offered for interoperability in a structured environment at a level of abstraction that is suitable to support the interoperability of environmental models in general. DRIHM is an appropriate driver and initial use case since it demands the handling of a wide range of hydro-meteorological models across meteorology, hydrology and hydraulics where the model coupling between these domains (not necessarily within the domains) is file-based and one-way.

## 2. Methods

We consider what would be necessary at a fundamental level to make a typical environmental numerical model interoperable with another in such a structured environment. It must be possible for a user to locate a numerical model of potential interest; it must be possible to evaluate the model for the targeted use, at least to a certain degree; it must be possible for the model to be set up and run either stand-alone or in concert with other linked numerical models; finally the user must then be able to interpret and perhaps visualise the results. For the specific use cases supported by the target DRIHM eInfrastructure, users must be able to discover and evaluate at least one of a meteorological model, an hydrological model or an hydraulic model that meets their spatial and temporal requirements as well as that of simulated phenomena; they must be able to compose a linear model chain crossing hydro-meteorological domains involving these models and then interrogate or visualise the results of each model in the chain. DRIHM also allows hydraulic model compositions (with two-way connections between models) as the final, downstream component.

Any such framework should be built on established concepts for model execution and interoperability and apply rigorous engineering methods and principles (emphasised, for example, by Wang et al., 2009). These concepts are apparent from standards and modelling systems which are already established with good track records. Two leading examples together exhibit the necessary characteristics, one standard from Europe and one modelling system from the USA:

- OpenMI, an accredited model interoperability standard from Europe which is generic in nature yet derived from the hydraulic modelling domain together with its FluidEarth implementation;
- the Community Surface Dynamics Modelling System (CSDMS) from the USA, promoting the modeling of earth surface processes, applicable across the geosciences and using integrated software models.

We abstract concepts embodied within these to formulate a generic framework which we then apply to the DRIHM eInfrastructure, also drawing from other related initiatives.

OpenMI (OGC OpenMI, 2014) is an accredited standard for model interoperability designed to enable the exchange of data between modelling components at run time. The first releases appeared in around 2004 with the latest version, 2.0 having been released in 2010. The specification for OpenMI consists of a core group of requirements and optional extensions. When satisfying the core requirements, a model becomes a 'Linkable Component' that can then be linked to other Linkable Components which also satisfy the core requirements. This Linkable Component would typically be a numerical model which can be run on its own or as an OpenMI composition of linked components. OpenMI includes requirements for describing components and the data they can exchange through qualitative or quantitative input and output 'Exchange Items'. The output exchange items refer to the outputs that a component offers to others and the input exchange items to the inputs that a component can validly accept from others. Automated semantic mediation between these Exchange Items is not part of the standard and quantities are defined by being broken down into their base dimensions. Although the most common use cases for applications of OpenMI involve time-stepping models, this aspect is not part of the core standard, but is offered in the Time-Space extension. The 'TimeHorizon' attribute provides the timeframe during which an exchange item will interact with other exchange items. Also, geometry can be represented as points, line segments, polylines, or polygons. The concept of 'Adaptors' is included in the standard to allow input and output exchange items to be pre or post processed in order to meet the requirements of other, linked models.

The FluidEarth Windows.Net implementation of OpenMI (Harpham et al., 2014) provides a software development kit (SDK) aiding the creation of OpenMI components together with a user